# Coded Aperture and Coded Exposure Photography: an investigation into applications and methods

Martin Wilson
WLSMAR028

Supervised by Dr Fred Nicolls

Department of Electrical Engineering
University of Cape Town

November 21, 2011



# Plagiarism Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination at this or any other university.

Signed :

by Martin Wilson at Cape Town.

# Abstract

This dissertation presents an introduction to the field of computational photography, and provides a survey of recent research. Specific attention is given to coded aperture and coded exposure theory and methods, as these form the basis for the experiments performed.

Coded apertures are optimized for the tasks of defocus deblurring and depth-from-defocus estimation, constructed using simple cardboard occluding masks, and then used to photograph test scenes in varying degrees of defocus. A sequence of coded aperture photographs is also used to capture a 4D light field with a standard SLR camera, and the captured light field is then used for stereo depth estimation and virtual refocusing applications. Furthermore, a coded exposure pattern is optimized for motion deblurring, and coded exposure photographs are captured of moving scenes by controlling the incident illumination.

The optimized coded apertures outperform a traditional circular aperture in both defocus deblurring and depth estimation tasks, and the optimized coded exposure patterns are shown to be superior to traditional box-shaped exposures in terms of motion deblurring accuracy. Finally, a meaningful stereo depth estimate is produced from the stationary light field camera, and visually convincing post-exposure refocused photographs are synthesized.

# Acknowledgements

*The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.*

*In addition, I would like to thank the following institutions and individuals for their assistance in the preparation of this thesis.*

*Armscor's PRISM program, managed by the CSIR, and the University of Cape Town (UCT) for their financial support.*

*My supervisor, Dr Fred Nicolls, who gave me the freedom to make mistakes, and the guidance to recover from them.*

*My parents, whose support and encouragement kept me motivated, even when the task seemed insurmountable.*

*My sister, Katie, who put up with my idiosyncrasies and bad grammer.*

*And lastly, my peers, who made the whole process enjoyable enough to forget that I was working.*

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Background & Problem Definition

Computational photography is a branch of research that utilizes computational imaging techniques in an attempt to extend or enhance the capabilities of digital photography. It is closely related to the more established fields of image processing, computer vision, and computer graphics, and it has recently become a highly active area of research, boasting a large number of participating research groups from around the world. Its popularity is due to the increasing availability of low-cost, high quality digital imaging equipment, as well as the numerous commercial and scientific applications for advancements in the field.

Currently digital imaging has applications in almost every area of industry and scientific research. However, the limitations of traditional photographic techniques and equipment are becoming increasingly apparent. Examples of such limitations include out-of-focus blurring, motion blurring, image noise, finite resolution, and the lack of depth information. All of these limitations reduce the usefulness and flexibility of digital photography. Therefore the problem posed to the computational photography researcher is to develop new and ingenious ways to eliminate or mitigate the effects of these limitations.

Fundamentally, computational photography calls for a paradigm shift from the traditional definition of photography to a far more general definition. Instead of merely capturing a 3D scene as a 2D intensity map, the ideal photograph should be able to capture additional information about the scene, which can then be used to build a more accurate, multi-dimensional representation of the scene's visual appearance. Coded photography techniques attempt to capture this additional visual information by reversibly encoding the optical signal before it is captured by the sensor. This thesis will investigate the advantages and disadvantages of two such coded photography techniques, namely, coded aperture photography and coded

exposure photography.

## 1.2 Objectives

The objectives of this research thesis are to:

1. Provide a brief introduction to the field of Computational Photography, and to investigate current research topics within the field.
2. Provide an overview of light field theory and methods, and to discuss examples of popular applications.
3. Research coded photography theory and methods in greater detail, with specific attention given to coded aperture and coded exposure techniques.
4. Design and construct a re-usable, reconfigurable coded photography prototype camera.
5. Capture a light field using the prototype camera, and to experiment with popular applications.
6. Experiment with coded apertures and coded exposures, giving specific attention to defocus deblurring, motion deblurring, and depth estimation.
7. Interpret the results of the experiments, and conclude on their successes and failures.
8. Provide a list of recommendations for future research that could extend the research scope, or improve performance.

## 1.3 Scope and Limitations

The scope of this thesis includes all recent computational photography techniques that can be used to extend and/or enhance the performance of optical camera systems, with emphasis on coded photography techniques. However, in order to focus the research presented and reduce ambiguity, the following limitations have been imposed:

- The focus of this thesis is on recent (post-2001) advancements in computational photography research. Older research will only be discussed if it is required to understand the newer techniques.
- In order to simplify the construction of the coded photography prototype camera, additions and modifications were made using only occluding optics. Therefore any methods that required novel refractive or reflective elements were not investigated.

- Only single-camera techniques were considered, and the motion of the camera during image capture was assumed to be negligible.
- The prototype camera had to consist of inexpensive, easily obtainable (non-specialist) components.
- Only still photography (i.e. no video) was considered in this thesis, and no attempt was made to make the processing real-time.
- Only grayscale images were used in the experiments. However, all the techniques covered in this thesis could be extended to color images by processing each color channel separately.
- All experiments were performed in carefully controlled laboratory conditions, and no real-world scenarios were tested.

## 1.4 Plan of Development

This thesis begins with a literature review (chapter 2), where current related research is surveyed and summarized. The literature review is intended to provide a background knowledge of the field of computational photography, and to give context to the experiments described later in the report. The specific topics that are reviewed include:

- Light field theory and methods.
  The concept of a light field is explained and number of geometric representations are described. Also, various methods for physically capturing a light field are presented and compared.
- Overview of computational photography research.
  Computational photography is a relatively new field of research that has developed out of the more traditional fields of computer vision, digital image processing, and optronics. The objectives of computational photography are discussed, and examples of interesting work conducted in the last decade are provided.
- Examples of novel illumination and novel camera research.
  All of the experiments performed in this report can be classified as either novel illumination or novel camera research, and therefore current work in these two categories is surveyed in greater detail.

In chapter 3, the design and construction of a prototype coded photography camera is detailed. The prototype camera allows for the use of non-standard aperture shapes and exposure patterns, and is essential to all the experiments described in this dissertation. Initially the goals and features of the prototype are listed, and the selection and interconnection of the

individual components are discussed. The chapter then describes the prototype's construction (including any relevant problems that were encountered), and finally the functionality of the prototype is tested and its performance analyzed.

Chapter 4 describes the theory and methods involved in coded aperture photography. The chapter begins by defining a geometric model of a lens-based camera, and describes how aperture effects can be modeled as a filter in a linear system. The rest of the chapter is devoted to discussing the specific implementation details regarding using coded apertures for defocus deblurring, estimating depth from defocus, and capturing light fields. In order to find optimal coded apertures for each of these tasks, performance metrics are devised and a genetic algorithm is described.

Chapter 5 describes the function of the shutter within the camera model, and explains its role in the formation of motion blur. The chapter then discuses the theory and methods of coded exposure photography, with specific attention given to how they can be used to reduce motion blur. Under certain conditions motion blur can also be modeled as a filter in a linear system, and therefore removing blur becomes a deconvolution problem. Finally, two performance metrics are devised for optimizing coded exposure patterns.

The experimental procedure is detailed in chapter 6. The chapter starts by defining the camera parameters used in the experiments and measuring the level of noise in the input images, and then moves on to discussing the individual implementation details for each experiment performed. The experiments include defocus deblurring and depth-from-defocus using coded apertures, stereo depth and virtual photograph synthesis using light fields, and motion deblurring using coded exposures.

The results of the experiments are displayed, analyzed and interpreted in chapter 7. In order to determine whether the additional complexity associated with coded photography methods is justified, the performances of the various coded aperture shapes and coded exposure patterns are compared with those of the traditional apertures and exposures.

Finally, in chapter 8 the results of the experiments are briefly summarized, and recommendations are made for future research that could extend the scope of this research project, or improve results.

# Chapter 2

# Literature Review

## 2.1 Light Fields

### 2.1.1 Overview

A *light field* is defined as the intensity of light rays in every direction, from every location in space [1]. Under the assumption of geometric optics (i.e. the effects of light acting as a particle can be safely ignored), the light ray is the elemental substance of light, and therefore a complete record of the all rays in a scene is the most fundamental representation of its visual information [1]. Due to the infinite number of light rays within any particular scene, a light field cannot be physically captured in its entirety, but is instead an idealized concept that remains useful for describing visual structure and analyzing potential observations [1]. The ideal light field can also be digitally sampled, and these partial light fields have been shown to be sufficient for applications such as synthesizing images from different points of view without any structural or geometric knowledge of the scene [12, 19].

The concept of the light field is not a new one, but was even noted by Leonardo Da Vinci when describing space being filled with an infinite number of "radiant pyramids," each one representing the view made by an observer at that point in space [13]. However, the recent increase in the availability of inexpensive, high-resolution cameras and customizable optical elements has made it easier to practically sample a light field, and this has rekindled research into the theory and design of light field cameras.

### 2.1.2 The 5D Plenoptic Function

The Plenoptic function gives the radiance along each light ray within a three-dimensional region of space [1]. A standard pinhole camera captures light rays passing through a single point in space (the aperture), and forms a 2D image based on their angle of entry. In this

way the pinhole image can be thought of as an intensity distribution at a single point in space with respect to the two angular directions, $\theta$ and $\phi$, in 3D space: $P(\theta, \phi)$. Therefore, one way to conceptualize the plenoptic function is as a function that can return the image formed by a pinhole camera placed at any point in the 3D scene. Figure 2.1 shows a partial plenoptic function defined by two ideal observers at distinct locations in a scene. The hypothetical pinhole cameras in this analogy differ from real pinhole cameras in that they can capture light rays from every angle (i.e. a 360° field-of-view).

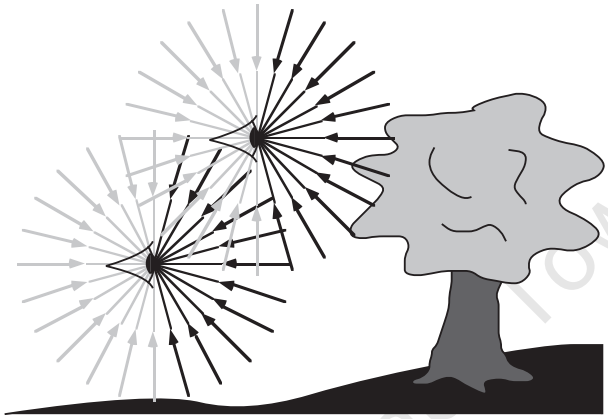


Figure 2.1: The 5D plenoptic function can be thought of as the view of a scene by observers at every possible location in space. Diagram taken from [1].

Since the pinhole cameras produce 2D representations of angular intensity at locations in 3D space, the entire plenoptic function becomes a function of five variables (5D) in the form:

$$Radiance = L(x, y, z, \theta, \phi) \tag{2.1}$$

where $x, y$, and $z$ represent a point in 3D space and $\theta$ and $\phi$ represent the angular direction of the light ray. This definition is shown geometrically in figure 2.2. The standard 5D plenoptic function is sometimes extended to include extra dimensions such as light ray frequency (color) and time, resulting in 6D and 7D versions respectively [1].

### 2.1.3 The 4D Light Field Function

In general, five dimensions are required to define the plenoptic function of a region in 3D space. However, when working in free-space (a region with perfectly transparent atmosphere and free of occluders) the radiance values remain constant along ray lines, and therefore 5D plenoptic functions of these regions will contain large amounts of redundancy. In fact, it can be shown that the plenoptic function in free-space contains exactly one dimension of redundancy [12].

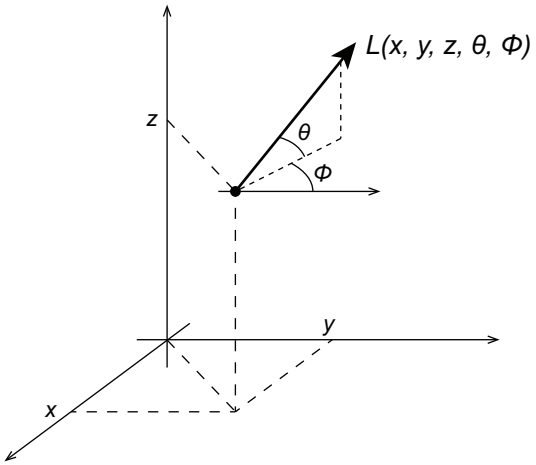



Figure 2.2: The 5D plenoptic function parameterized by 3 spatial dimensions and 2 angular dimensions.

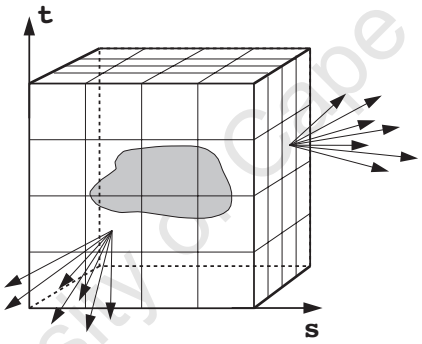



Figure 2.3: If a scene can be completely encompassed by a convex surface (e.g. a cube), then the values of the 5D plenoptic function outside this region can be calculated directly from the values on the surface.



Figure 2.3 shows a scene in three-dimensional free-space, completely encompassed by a convex surface (i.e. a cube). It should be clear that once the light rays originating from the scene pass through the surface of the cube they will remain constant indefinitely. Therefore, a ray passing through any point outside the cube can be traced back to a point on the surface of the cube, and its radiance value will remain unchanged. In this way, the plenoptic function need only be defined on the surface of the cube, and since a point on a surface can be defined by only two variables (compared to the three variable required in 3D space), the 5D plenoptic function can be reduced to a 4D function (two spatial dimensions and two angular dimensions) [12]. This 4D function is referred to as a *light field* [19] or a *lumigraph* [12].

While it might seem limiting to restrict the light field model to convex surfaces in free-space, in practice these assumptions are quite acceptable. Specifically, concave scenes can easily be handled by restricting the analysis to outside its convex hull, and where the observers must be located within the convex hull (such as architectural scenes) then the viewer can be enclosed in the bounding surface. The reduction from five to four dimensions also brings with it advantages that far outweigh the limitations. For example, the 4D light field is easier to physically measure, decreases the size of the entire dataset, and greatly simplifies the mathematical operations involved [19].

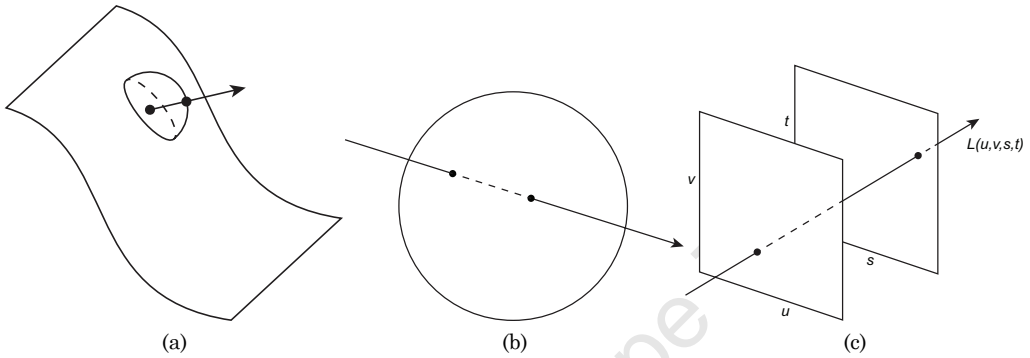



Figure 2.4: Three different parameterizations of the 4D light field: a) position on a surface (2D) and direction from normal (2D), b) entrance and exit points on a sphere (2 × 2D), and c) the intersection points on two planes (2 × 2D). Images inspired by [19] and [12].

There are many ways to parameterize the four dimensions of a light field, and each parameterization suits a different application. Figure 2.4 shows three of the most popular parameterizations: (a) location and direction on a surface, (b) entrance and exit points on a sphere, and (c) the intersection points on two planes. The two plane parameterization (which is commonly referred to as a *light slab*) is by far the most common due to its rectilinear coordinates and because calculating ray projections involves only linear algebra (compared to the trigonometric functions required by parameterizations that use angular variables) [19]. Unless stated otherwise, all 4D light fields in this document will be parameterized in this way:

$$Radiance = L(u, v, s, t) \tag{2.2}$$

where $(u, v)$ and $(s, t)$ are the 2D coordinates on each plane respectively.

Once a light field is obtained, images can be synthesized by taking 2D 'slices' from the 4D function. Figure 2.5 shows how this is achieved by placing a virtual image plane and observer into the model. Rays are projected from the observer through the image plane, their

intersections with the planes in the light slab are calculated, and finally the corresponding intensity values are determined by resampling. While the simple method described here produces pinhole images, real cameras with finite apertures and lens distortions can also be synthesized from the light field [12].

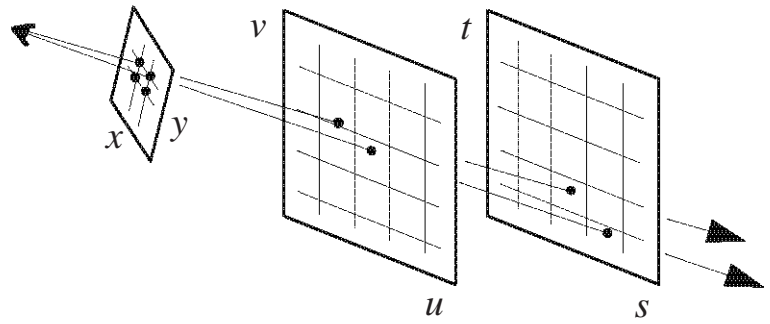



Figure 2.5: 2D Images can be synthesized from the 4D light field by projecting the rays onto a virtual image plane. Image taken from [19].

### 2.1.4 Capturing Light Fields

The first step in capturing a light field is to choose a sampling pattern. The sampling pattern needs to suit both the scene and the intended application, as it will determine important experimental parameters such as spacial resolution, angular resolution and risk of aliasing. The second step is to measure the radiance level for each ray in the chosen sampling pattern. Measurements could be performed naïvely by using a spot radiometer aimed in the direction of the target ray, but for any practical application (which can contain thousands or millions of rays) this is prohibitively tedious. A far more efficient method is to use arrays of 2D photographs [19].

Figure 2.6 shows two ways in which a light field can be visualized using a 2D array of 2D images. In (a) each image in the array represents a single point on the $(u, v)$ plane, and each pixel in the image represents a point on the $(s, t)$ plane. This is similar to the analogy presented earlier that a light field can be thought of as an infinite collection of pinhole cameras, since each point on the $(u, v)$ plane can thought of as a camera. The second way, shown in (b), is merely the reverse of (a). Each image in the array now represents a single point on the $(s, t)$ plane while each pixel represents a point on the $(u, v)$ plane, and results in images that are predictably reminiscent of reflectance maps. The two methods are mathematically identical, and a light field can be converted from one visualization to the other by simple pixel rearrangement [19].

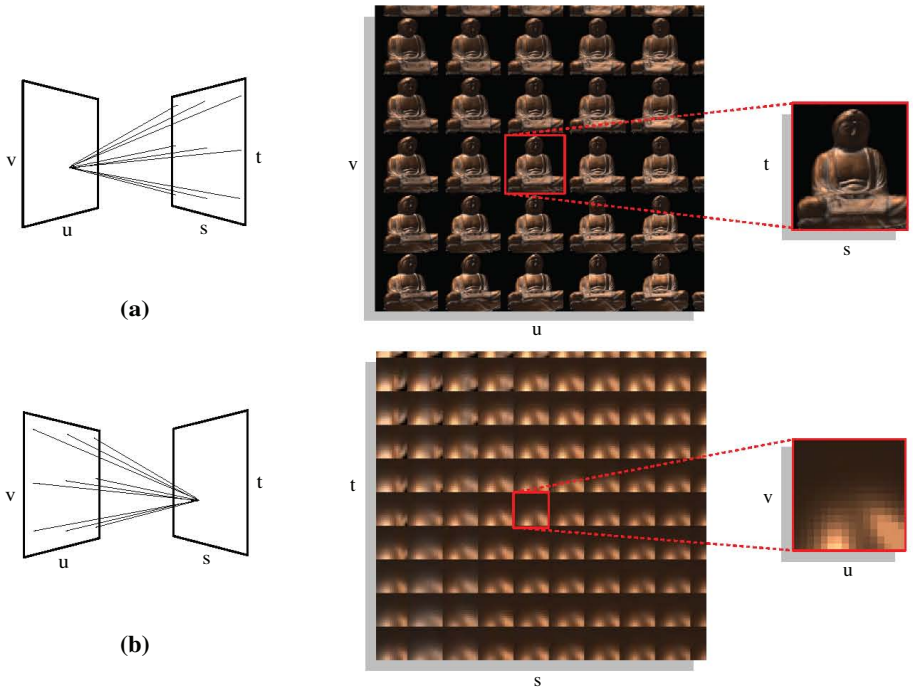



Figure 2.6: Two ways to visualize a light field as collection of images: in (a) each image in the array represents a single point on the $(u, v)$ plane and each pixel represents a different point on the $(s, t)$ plane, while in (b) the converse is true. Image taken from [19].

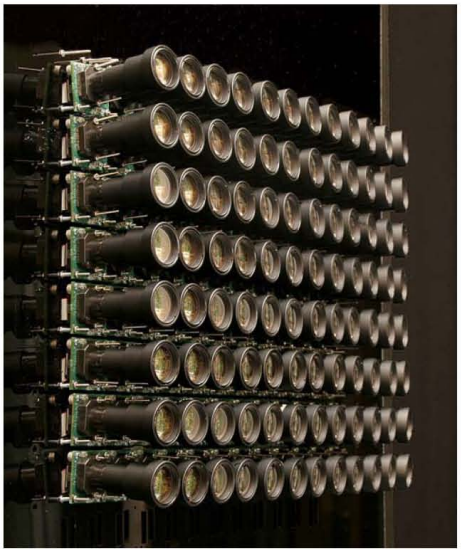


Figure 2.7: A light field camera built from an array of tightly packed cameras. Image taken from [38].

The simplest way to capture an array of 2D images of a scene is to use an array of cameras, and this was the method used by early light field researchers. The camera array built by Bennett Wilburn et al. at Stanford University in 2005 (shown in figure 2.7) is a good example of this type of light field camera. Camera arrays capture the light field as shown in figure 2.6 (a), where the captured photograph from each camera in the array represents a single point on the $(u, v)$ plane. Despite their conceptual simplicity, camera arrays are difficult to use in practice due to their large size and reliance on accurate camera orientation calibration.

Capturing a light field with a single camera is far more portable and robust than using a camera array or gantry, and therefore building such systems is currently a popular area of research. In order to capture a light field with a single camera, additional optical elements need to be inserted into the camera between the sensor and the lens that allow the measurement of a ray's angular direction. In all cases, this is achieved as a trade-off with some other quantity such as spacial resolution or dynamic range. The camera developed by Ren Ng et al. [27] at Stanford University in 2005 uses a micro-lens array inserted into a standard Single Lens Reflex (SLR) camera between the sensor and the lens (as shown in figure 2.8). Each micro-lens disperses the rays passing through its center onto a select number of pixels on the sensor, thereby forming arrays of discrete circular regions in its output (see figure 2.9). Conceptually the micro-lens array represents the $(s, t)$ plane while the sensor represents the $(u, v)$ plane, and therefore the tiled output visualizes the light field in the manner described in figure 2.6 (b), where each circular region represents a single spacial point on the micro-lens array.

Not all light field cameras require additional refractive elements. For example, the camera developed by Veeraraghavan et al. [36] in 2007 uses only an attenuating mask between the sensor and the lens to measure the 4D light field. Their attenuating mask contains a high frequency sinusoidal pattern and this is multiplied with the incoming light rays to create spectral tiles of the 4D light field in a method analogous to *heterodyning* in radio electronics. Not only is the attenuating mask less expensive and easier to produce than a micro-lens array, but it also modulates the light signal in a reversible way, which means that the full resolution 2D image can be recovered when the 4D light field is not required [36].

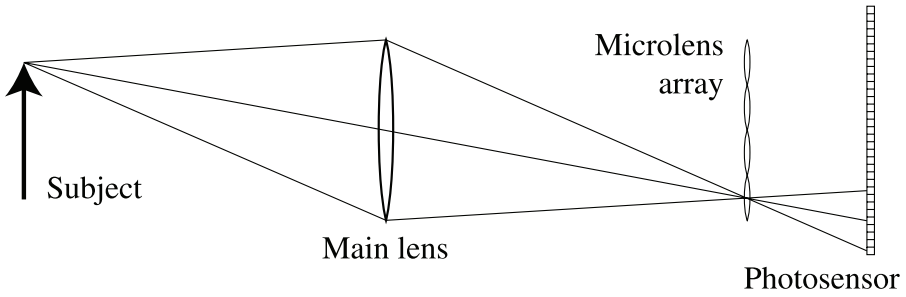



Figure 2.8: A schematic of a light field camera built by adding a micro-lens array between the sensor and the main lens. Image taken from [27].



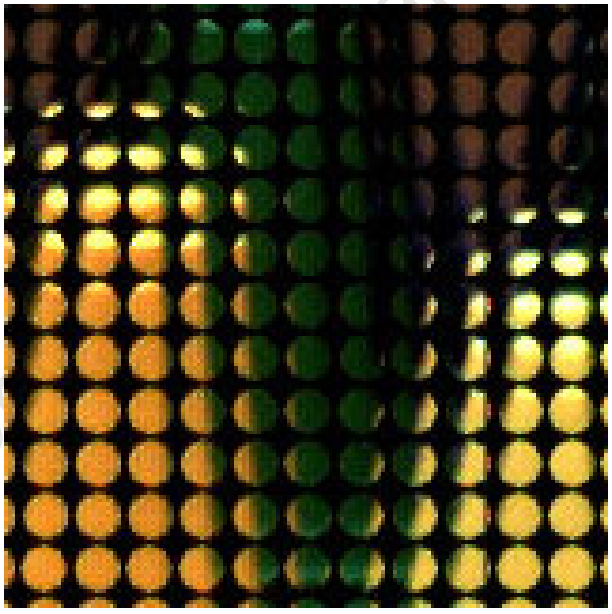


Figure 2.9: A sample of raw output from a light field camera using a micro-lens array. Image taken from [27].

## 2.2 Computational Photography

### 2.2.1 Overview

Computational photography attempts to extend or enhance the capabilities of digital photography by incorporating computational elements into the traditional photographic model. Furthermore, computational photography also aims to push the boundaries of what we understand photography to be, by capturing not only 2D light intensities, but rather the entire machine-readable visual experience [6]. Often these goals are achieved by modulating (or *coding*) the light field at various stages in the photographic model according to an engineered sequence or pattern, and these methods are collectively referred to as *coded photography*.

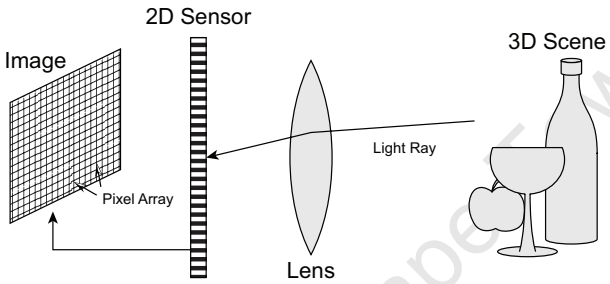



Figure 2.10: Diagram showing the traditional film-like photographic model. Diagram inspired from [6].

Figure 2.10 shows the traditional film-like photographic model, which has remained largely unchanged (despite the conversion from chemical to electronic sensors) since its discovery at beginning of the nineteenth century. Light rays originating from a 3D physical scene are bent through a lens which focuses the rays onto a 2D image sensor. The instantaneous light intensity values are measured at discrete locations on the sensor, and are aggregated together to form a digital image. However, this conversion from 3D scene to 2D intensity array discards vast amounts of useful information such as illumination structure, geometric shape, motion and surface reflectance properties. In order to capture some of this lost information, the computational photography model calls for the generalization of each element in the model through the use of computational elements [6].

The generalized computational photography model is shown in figure 2.11. Instead of focussing on 2D intensity values, this model attempts to record the paths of light rays from their origin at the illumination source all the way through to reaching the eyes of the viewer. The rays from the light sources first pass through generalized modulation and optical stages in order to produce a specifically engineered 4D incident light field, and are then modulated by the scene to form the 4D reflected light field. In this way the scene operates as a 8D

modulation function that completely defines the scene's optical properties. The reflected rays then pass through generalized optics in the camera, fall onto a multidimensional sensor, and are converted and processed as information signals. Finally, a true photographic display would take the processed information signals and recreate the 4D light field for the viewer. Almost all computational photography research can be categorized into one (or more) of the three subsections in the model, namely, novel Illumination, novel cameras, and novel displays [6].

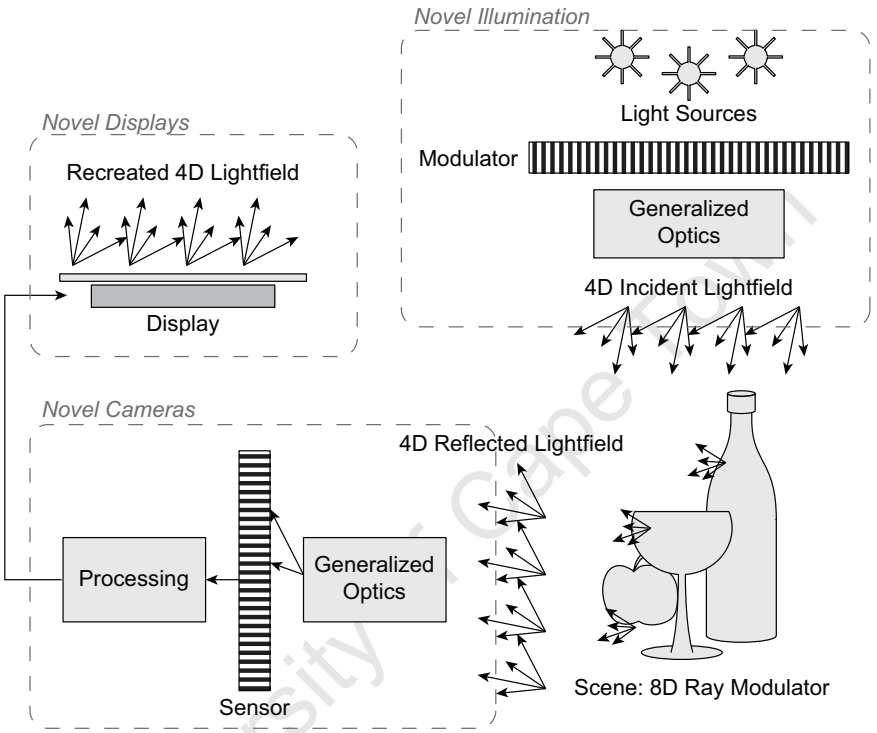



Figure 2.11: Diagram showing the major elements within the field of computational photography research. Diagram inspired from [6].

### 2.2.2 Novel Illumination

Novel illumination refers to any technique that attempts to specifically engineer the 4D incident light field before being reflected by the scene. By adding structure to the incident light and then comparing it with the captured light reflected from the scene, one can accurately deduce various physical attributes of the scene. Recent research has shown that it is possible to capture properties as diverse as depth [22] and depth discontinuities [30], post-capture relighting information [32, 37], bidirectional reflectance distribution functions (BRDFs) [37, 11], lighting composition (diffuse vs. specular, or direct vs. global) [16, 25], and even the capture of images without a camera [33].

While many of the properties mentioned above can be captured using more traditional methods, novel illumination techniques usually require less specialized hardware, are more adaptable, and can be easily retrofitted into existing photographic systems. Novel illumination owes these advantages to the recent availability of accurate but inexpensive light-sources that can be electronically controlled and modulated. Examples of such light-sources are multiple synchronized electronic flash modules [30], high-frequency LED based light arrays [37], and video projectors [22, 32, 11, 16, 25, 33]. An example of using a video projector to create structured illumination can be seen in figure 2.12.

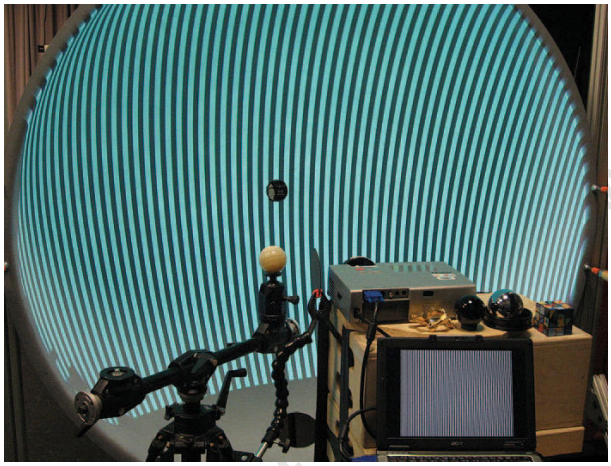


Figure 2.12: An example of structured illumination created using a video projector and a hemispheric dome reflector [16].

### 2.2.3 Novel Cameras

The two main components of the Computational Photography camera are the generalized optics that focus the incoming light rays, and the image sensor that converts the rays into digital signals. Recently, novel cameras have experimented with unconventional optics and sensors in order to improve the camera's performance and to capture additional information about the scene.

One advantage of using unconventional optics is the ability to engineer the camera's point-spread function (PSF). The PSF of a camera is defined as the output obtained from the camera when a point light source is the input, and it is a measure of how accurately the camera maps points in the scene to points in the image. The ideal PSF would be a point itself, however all real cameras have non-ideal PSFs and this is the fundamental cause of blurring in images. One popular approach to reducing out-of-focus blur is to engineer the PSF to be depth invariant using wavefront coding. Wavefront coding uses a phase-plate

with spatially varying diffraction index (or thickness) to adjust the phase of the light rays passing through it [8]. Another approach is to use a coded aperture to engineer the shape of the PSF in such a way as to conserve high-frequency content, thereby increasing the performance of deblurring using deconvolution [18, 36]. Extending the notion of the PSF into the time domain is useful for describing motion blur in an image, and these time-orientated PSFs can be engineered in similar ways. For example, modulating the incoming rays in the time domain using a coded shutter greatly improves one's ability to reduce motion-blur caused by a moving subject [29].

Unconventional optic configurations using both reflective and refractive elements can also be used to engineer the field-of-view of a camera, allowing the capture of traditionally impossible photographs. For example, radial catadioptric cameras use cylindrical reflective elements to view a scene from multiple perspectives in a single photograph. This multi-view image can then be used to estimate properties such as scene depth (as shown in figure 2.13) and BRDF [15]. In some cases the geometry of the camera optics is so far removed from traditional camera structures (and hence from the structure of the human eye), that the output is completely unintelligible without first going through post-capture processing. For example, the output of lenses made entirely from reflective elements (without any refractive elements) appear at first to be completely random, however accurate images can still be achieved by using image processing and machine learning techniques. In addition to being less expensive than traditional lenses, these methods also hold potential for super-resolution and 3D imaging [35, 9].

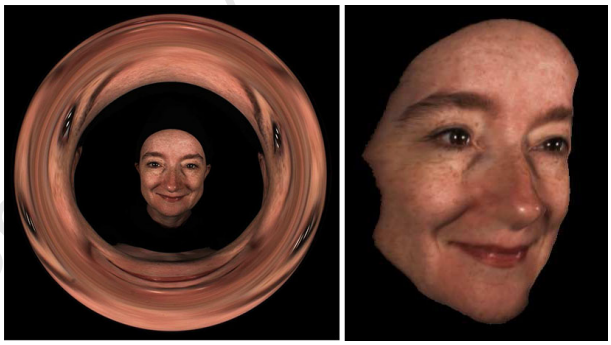


Figure 2.13: The output from a radial catadioptric camera (left), and the 3D surface estimate obtained from it (right) [15].

The traditional camera sensor can be described as a 2D array of light-meters, and while this is the most obvious configuration, it is not necessarily optimal for all applications. Recently the traditional design ideals of small, non-overlapping rectilinear pixels, capturing only intensity information, and being sampled simultaneously at a single instance in time are being

challenged. Research has shown that non-periodic and non-rectilinear pixel layouts provide several advantages in applications such as super-resolution [3], and that non-uniform filter mosaics can be used to improve color sensitivity and dynamic range [26]. The image sensor is also no longer limited to measuring just intensity values, as 4D light field sensors have been created using micro-lens arrays [27] and coded amplitude masks [36]. Lastly, even the static, rigid format of the image sensor is being challenged by creating flexible array sensors (as seen in figure 2.14) [31, 28], and by using micro-actuators to adjust the position of the sensor within the camera during image capture [23].

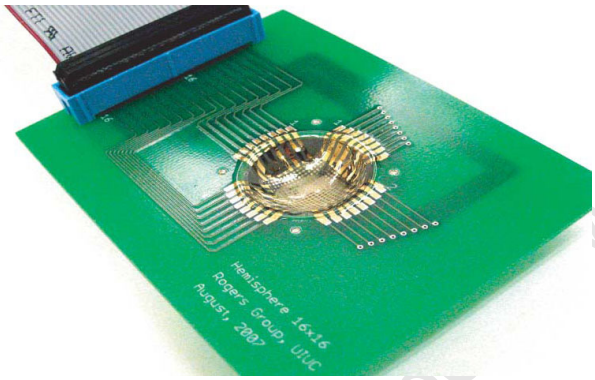



Figure 2.14: A prototype hemispherical image sensor. Photodetectors and flexible interconnects mounted onto molded silicone [31].

### 2.2.4 Novel Displays

The last stage of the generalized photographic model is the novel display. The ideal display would be able to take the captured and processed light field from the camera and reconstruct it exactly for viewing by the user. However a number of challenges make this goal difficult to achieve in practice [6]. For example, user displays are inherently real time and therefore time-multiplexing (popular in novel illumination and cameras) cannot be employed to reduce the high-bandwidth required by a 4D light field. Furthermore, displays must contend with multiple environmental conditions, including but not limited to, user viewing angle, multiple viewers, ambient lighting conditions and space limitations [6].

Current research into novel displays can be divided into the following categories: virtual (or augmented) reality systems [2], stereoscopic (and autostereoscopic) displays [7], holography [4], and volumetric displays [4]. Some of these systems require the user to wear specialized equipment such as head-mounted displays or filtering glasses, while others attempt to reconstruct the light field at the source by using combinations of projectors, microlens arrays, and other unconventional optical elements and materials [4].

# Chapter 3

# Design and Construction of a Coded Photography Camera



## 3.1 Design

### 3.1.1 Goals and Features

The primary goal of the prototype camera was to provide the coded photography researcher with a simple and flexible tool that could be reconfigured to perform a large variety of coded aperture and coded exposure experiments. The camera was designed to include useful parameter automation wherever possible, while still remaining robust enough to repeatably capture accurate results. The design of our prototype camera was influenced by the construction of similar cameras by other researchers [18, 29], and we have merely attempted to collate each of their individual design elements into a single system.

The complete list of required features is shown below:

- Ability to capture high-resolution still images, with high signal-to-noise ratios.
- Reconfigurable aperture, allowing for a large variety of shapes and sizes.
- Programmable illumination patterns, of variable length and duration.
- Easy selection of camera parameters (shutter-speed, sensor gain, focus, etc.).
- Non-permanent construction to allow for easy component removal and replacement.
- Constructed from low-cost, easily obtainable components.
- Clear and documented interface, allowing future use by other researchers.

### 3.1.2 System Overview

Figure 3.1 shows a block-diagram overview of the prototype camera's components and their interconnection. Each component in the system is described in more detail below.

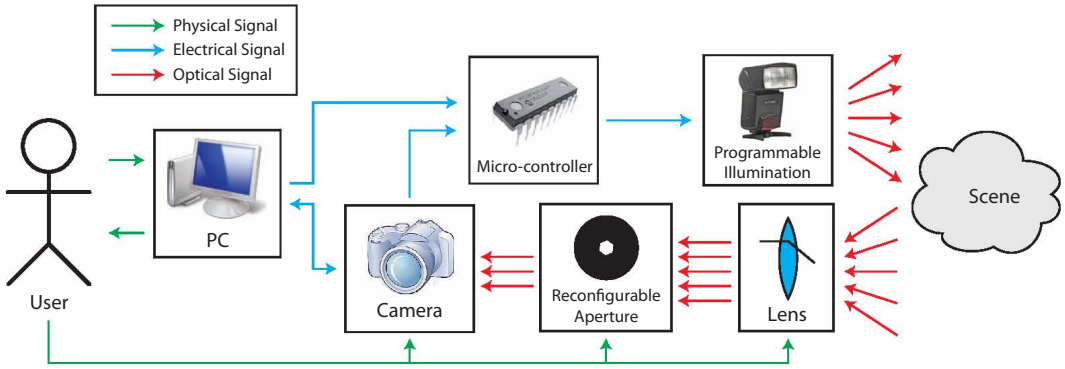



Figure 3.1: A block-diagram overview of the prototype camera, showing major components.

#### Personal Computer

The personal computer (PC) is the main interface between the user and the rest of the system. The user can use the PC to configure the camera parameters (e.g. shutter speed, ISO level, etc.) and to set the chosen illumination pattern, as well as to receive visual output from the camera. The PC is also responsible for performing any image-processing and scheduling tasks that may be required. The PC is connected to the camera and the illumination micro-controller via the Universal Serial Bus (USB) interface.

The minimum requirements for the PC were relatively basic, and are listed below:

- graphical user interface (GUI).
- CPU sufficient for standard image-processing tasks.
- >10GB of free hard-drive space for parameter and image data storage.
- Universal Serial Bus (USB) interface.

#### Camera

The camera performs the actual image acquisition, and includes an image sensor, mechanical shutter, and associated electronics. The camera receives the optical rays that pass through the lens and aperture, and converts them into an electrical signal. Some camera parameters cannot be configured by the PC and so must be set physically (e.g. orientation,

manual-focus, and advanced menu options). In these situations the user will have to physically interact with the camera.

Many coded-aperture techniques encode additional information into the image at the expense of spatial resolution. Therefore it is vital that the camera's sensor be able to provide enough initial resolution to make these techniques feasible. Also, since the reconfigurable aperture lies between the sensor and the lens, it must be possible to detach the lens from the camera body.

The minimum requirements for the camera are listed below:

- USB interface, with the ability to be controlled remotely from the PC.
- Interchangeable lens mount.
- High resolution sensor ( >8 Megapixels).
- Ability to accurately set each camera parameter (shutter-speed, sensor gain, etc.).
- Tripod mount for robust positioning.

**Reconfigurable Aperture**

The reconfigurable aperture replaces the traditional hexagonal aperture in the lens, allowing the researcher to experiment with a large variety of aperture shapes. For simplicity and ease of construction the reconfigurable aperture was restricted to shapes with fully transparent and fully opaque areas (i.e. binary), and took the form of an $N \times N$ grid of blocks (allowing $2^N$ different aperture shapes). Selecting the value of $N$ is non-trivial, and is discussed later in the aperture selection section.

In order to allow for the practical comparison of large numbers of aperture shapes, reconfiguring the aperture needs to be a quick and simple task. For example, the researcher should be able to reconfigure the aperture without dismantling the lens or affecting the orientation of the camera. In order to achieve this goal the coded aperture was designed to be inserted into the lens through a slit in the outer body of the lens. The other requirements for the reconfigurable aperture are listed below:

- Physical dimensions must allow for its easy insertion into the chosen lens.
- High contrast between opaque and transparent areas.
- Medium to high resolution to allow for many different aperture shapes.

**Lens**

The lens takes incident optical rays from the scene and bends them appropriately to form an image of the scene on the camera's sensor. The aims of this prototype camera did not include investigating lens-specific phenomenon, and therefore the selection of the lens was not vital. However, the lens did need to be compatible with the selected camera's lens mount, and had to be able to be manually focused.

An off-the-shelf lens includes delicate aperture leaves that provide the user with a resizable, approximately circular internal aperture. This internal aperture needed to be removed and replaced with the reconfigurable aperture, and therefore it was likely that the lens would become irreversibly modified in the process. For this reason, ease of disassembly and cost become important considerations. Lastly, the reconfigurable aperture had to fit into the space occupied by the old internal aperture, and so a lens with a large maximum aperture was preferred.

The minimum requirements for the lens are listed below:

- Inexpensive and simple (without complicated systems such as zoom and image-stabilization).
- Camera-compatible mount.
- Manually focusable.
- Inexpensive, since it would be irreversibly modified.
- Large maximum aperture.

**Micro-Controller**

The micro-controller is responsible for triggering the coded illumination according to the timing pattern supplied via the PC. The timing has to be highly accurate (down to microsecond level) in order to achieve repeatable results and to allow the use of moderately complex patterns within the relatively short exposure time of the camera. This high accuracy is achieved by preloading the timing pattern into the micro-controller via the USB connection and then using the high-speed flash-sync signal from the camera to trigger the micro-controller's output signal.

The basic requirements for the micro-controller are listed below:

- USB serial interface.
- At least one free digital output pin.

- At least one free external interrupt pin.
- A microsecond-accurate internal timer.
- A small amount of non-volatile data storage.

**Programmable Illumination**

The programmable illumination module simply converts the coded electrical signal from the micro-controller into an optical signal, therefore the main electrical requirement is that the module be capable of switching at the same speed as the incoming signal. Also, since the programmable illumination is used for coded exposure experiments, the module must be powerful enough to comfortably overpower the ambient environmental lighting and must illuminate the entire test scene without producing strong shadows or reflections. This means that the programmable illumination should be as close as possible to the test scene and that it should illuminate the scene from many directions.

The requirements for the programmable illumination are listed below:

- Switching speed should be equal to or greater than the maximum speed of the input signal.
- Maximum optical power level should comfortably exceed that of the ambient environmental lighting.
- Must illuminate the test scene uniformly and from multiple directions.

### 3.1.3 Component Selection

**Personal Computer**

The PCs in the UCT Digital Image Processing lab already met all the required specifications, and therefore no new equipment was required. The PC that was used for all data capture and processing had the following specifications: Intel Core i5 2.67 GHz quad-core processor, 8GB of RAM, running Matlab R2010a (64-bit) on the Mac OS X (10.6.6) operating system.

**Camera**

The consumer digital cameras that were commercially available at the time could be sorted into three general categories: compact point-and-shoot, single-lens reflex (SLR), and professional-grade medium format. These three formats are compared in the table below according to their features.

| | Point-and-Shoot | SLR | Medium Format |
|---|---|---|---|
| **Price (approx.)** | R 2,000 - R 5,000 | R 5,000 - R 8,000 | R 100,000 + |
| **Resolution** | 8 - 12 MP | 10 - 15 MP | 22+ MP |
| **Interchangeable Lens** | No | Yes | Yes |
| **Manual Controls** | No | Yes | Yes |
| **Remote Control** | Some | Yes | Yes |
| **Mechanical Shutter** | No | Yes | Yes |
| **Access to Sensor** | None | Limited | Full |

The compact point-and-shoot camera format was clearly the least expensive option. However, many of the required features listed earlier were not provided, such as interchangeable lenses and manual controls. This fact made point-and-shoot cameras highly unsuitable for coded photography research.

The SLR camera format provided all the required features with only a slightly higher price than the point-and-shoot cameras. The flexibility provided by the SLR format will also allow the camera to be used in a wider variety of future research applications. The only feature that the SLR lacked was full access to the camera's sensor, and while this may have been useful in some tasks, most Coded Photography techniques can be implemented without it.

Lastly, the professional-grade, medium format cameras provided all the required features, as well as many other extras. Features such as resolutions of over 22 Megapixels and full access to the camera sensor would have been useful to have, but did they not justify the enormous price associated with these cameras. Therefore the SLR format was chosen for the prototype camera.

Currently the South African Digital SLR market is dominated by two manufacturers: Canon and Nikon. Cameras, accessories, and support from these manufacturers are widely available, and they are compatible with almost all third-party hardware and software. The two manufacturers offer similar camera models at comparable prices, and the choice between the two is predominantly based on user preference. The research student who designed the prototype camera had over five years of practical experience with a variety of Canon's digital SLR models, and therefore in order to make efficient use of this experience only cameras manufactured by Canon were considered.

Canon offered a variety of digital SLR models at various price-points. The table below compares the main specifications of four of their 'prosumer' (consumer/professional) models that were popular at the time:

| | Canon 1000D | Canon 500D | Canon 550D | Canon 50D |
|---|---|---|---|---|
| **Price (approx.)** | R 4,000 | R 6,000 | R 7,500 | R 8,500 |
| **Date Announced** | 06/2008 | 03/2009 | 02/2010 | 08/2008 |
| **Resolution (MP)** | 10.1 | 15.1 | 18 | 15.1 |
| **Material** | Plastic | Plastic | Plastic | Mg Alloy |
| **LCD Size** | 2.5" | 3.0" | 3.0" | 3.0" |
| **Cont. Capture** | 1.5 fps | 3.4 fps | 3.7 fps | 6.3 fps |

From the table, a general trend can be seen where the newer the camera, the higher the resolution and the higher the price. The only model that did not fit this trend was the Canon 50D, which is aimed at the professional photographer and therefore has a sturdier construction and faster response-time. Since the prototype camera was intended for indoor laboratory use only, the Canon 50D's robust Magnesium-alloy construction was not a major advantage, and was therefore not worth the extra cost.

The fundamental difference between the other three models was resolution, which was almost directly proportional to their price (≈R400 per megapixel). While a higher resolution is advantageous in coded aperture research, it is certainly not directly proportional (i.e. a 50% increase in resolution does not deliver results that are 50% more meaningful). Therefore, since the 500D and 550D differed only in resolution, we eliminated the 550D from consideration.

Resolution was again the most significant difference between the 1000D and the 500D, but there were also other, more subtle differences which had to be taken into account. These included the larger LCD display of the 500D, its faster continuous capture rate, and its superior auto-focus system. While none of these improvements were individually vital to the success of the prototype system, they all served to increase the value-for-money of the 500D, and made it a more useful and versatile device. For this reason the Canon 500D was selected as the optimal device for use in this prototype.



**Lens**

According to the list of requirements discussed earlier, the lens used in this prototype needed to be inexpensive, manually focusable, and have a large maximum aperture. Fortunately, Canon's most inexpensive lens at the time, the Canon EF 50mm f/1.8 II, met all of these requirements. The lens's specifications are shown in the following table.

| | |
|---|---|
| **Focal Length** | 50mm |
| **Number of Elements** | 6 |
| **Aperture Range** | f/1.8-16 |
| **Min. Focus Dist.** | 45 cm |
| **Weight** | 130g |
| **Price (approx.)** | R 1000 |

The Canon EF 50mm f/1.8 II had a simple design and construction, and this allowed for its easy disassembly. The lens's high quality output and low price-tag had also made it Canon's most popular lens at the time, and therefore there was a vast amount of freely available manufacturer and user information. For these reasons, the Canon EF 50mm f/1.8 II was the obvious choice for our prototype.

**Micro-Controller**

The micro-controller requirements listed in the previous section are relatively basic, and so the choice of micro-controller is based largely on convenience and personal preference. However, support for a USB serial connection is usually not supported on-chip, and requires additional computational components. For this reason it was decided to use a ready-made micro-controller development board with USB support already included.

The popular open-source Arduino Uno prototyping board (based on the Atmel ATmega family of micro-controllers [5]) was selected for use in our prototype system due to its wide user support-base, inexpensive price, ample computational resources, and its vast array of input-output connections. The Arduino Uno's relevant technical specifications are listed below:

| | |
|---|---|
| **Micro-controller Chip** | Atmel ATmega328 [5] |
| **Operating Voltage** | 5v |
| **Number of IO Pins** | 14 |
| **Flash Memory** | 32 KB |
| **Clock Speed** | 16 MHz |

## 3.2 Construction

### 3.2.1 Coded Aperture

The first step in constructing the reconfigurable aperture was to disassemble the lens in order to access the aperture plane, which was located between the front and rear lens elements. The lens was photographed at each step in the disassembly, and every component was measured and documented in order to create a detailed technical description of the unaltered lens. The photographs and technical drawings produced can be found in appendix B.


Figure 3.2: A photograph showing the particular stage in the lens disassembly where the front optical element was separated from the aperture module.

The next step in the construction was to free up enough space within the lens to allow for the insertion of the reconfigurable aperture. To achieve this, both the standard aperture module as well as the autofocus module had to be removed. This involved removing the aperture leaves and associated mechanisms, and desoldering the auto-focus circuit-board and motor. These elements are shown in figure 3.3.

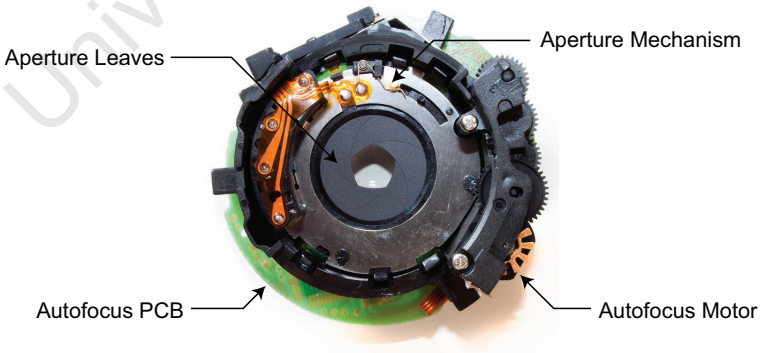



Figure 3.3: Diagram showing the various elements that had to be removed from the lens in order to create space for the reconfigurable aperture.

The simplest method for changing the aperture shape quickly was to place all the required shapes onto a long strip of mask material, and then feed this array of shapes into the lens through a slit in the outer body of the lens. Figure 3.4 shows how a strip of apertures was inserted into the lens. Inside the lens the strip of aperture shapes needed to be held securely at the aperture plane, and therefore a plastic brace was designed that would accept the aperture strip and guide it to the correct position within the lens. Figure 3.5 is a diagram of the aperture brace, and shows how it secures the strip of aperture masks. The brace was cut from a flat sheet of polyester plastic and folded into shape. Figure 3.6 shows the constructed aperture brace performing its function within the lens. Detailed design information regarding the aperture brace can be found in appendix B.

The final part of the reconfigurable aperture was the aperture mask itself. Various methods and materials were experimented with, including printing the designs onto overhead transparencies and developing the designs onto photographic film negatives. However, neither of these methods produced acceptable contrast between the opaque and transparent areas of the mask. Instead, the method that produced the best results was manually cutting the desired aperture shape out of an opaque medium such as cardboard. While this method did prevent the use of shapes that had opaque areas completely surrounded by transparent areas (such as concentric circles), its simplicity and low cost outweighed this small disadvantage. Figure 3.7 shows an aperture mask constructed in this manner.


Figure 3.4: A photograph of the modified lens, showing the slots cut into the body to allow the aperture strip to be inserted.

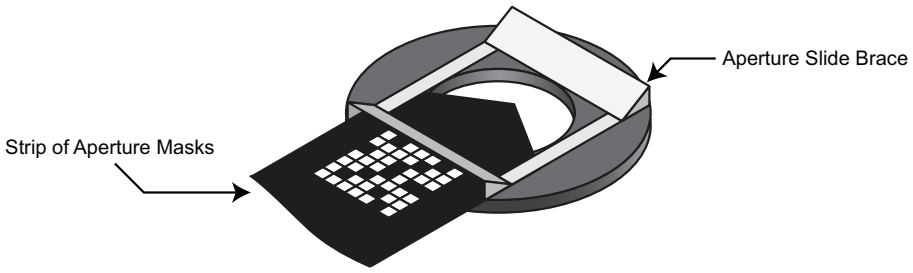



Figure 3.5: Diagram showing the design of the aperture slide brace that receives the strip of aperture masks and secures it to the aperture plane.

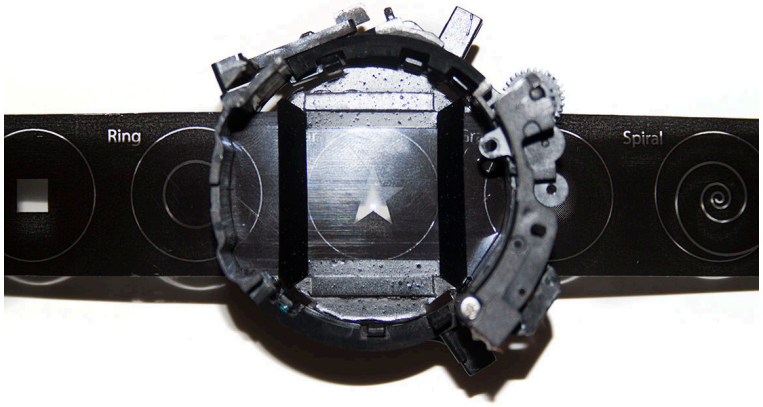



Figure 3.6: A photograph showing the constructed aperture slide brace inserted into the aperture plane of the lens.

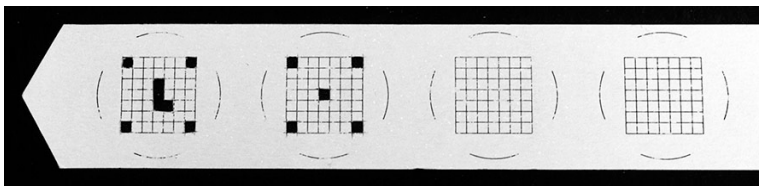


Figure 3.7: A photograph showing aperture masks constructed by cutting the desired shape out of a strip of cardboard.

### 3.2.2 Coded Illumination

As mentioned earlier, in order to use moderately long illumination patterns within a single exposure time, the programmable illumination module needed to be able to switch on and off at a high speed. Another requirement was that the programmable illumination needed to be powerful enough to comfortably overpower the ambient lighting in the scene. In order to satisfy these two requirements the prototype was constructed using an array of 48 white LED light sources. The array was constructed in a circular ring shape so that it could surround the camera's lens, thereby illuminating the scene from many different directions and eliminating problematic shadows. Figure 3.8 shows a photo of the constructed LED array.

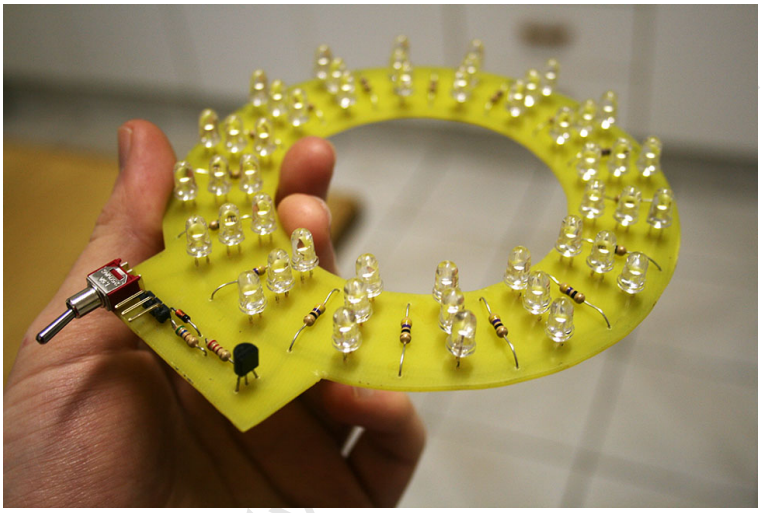


Figure 3.8: A photograph showing the constructed circular LED light array.



In order to time the coded illumination pattern accurately, the micro-controller needed to be able to detect when the camera shutter was opened. The simplest method for achieving this was to route the flash-sync signal provided by the camera's hot-shoe connector to one of the micro-controller's external interrupt pins. Each time this interrupt was triggered the micro-controller waited a user-definable initial delay, and then pulsed one of its output pins according to the pre-stored flash pattern and interval delay. The output of the micro-controller was then used to power the LED array through a transistor. The simplified connection diagram for the coded illumination module is shown in figure 3.9, and a flow-chart of its operation as well as some example timing diagrams are shown in figure 3.10.

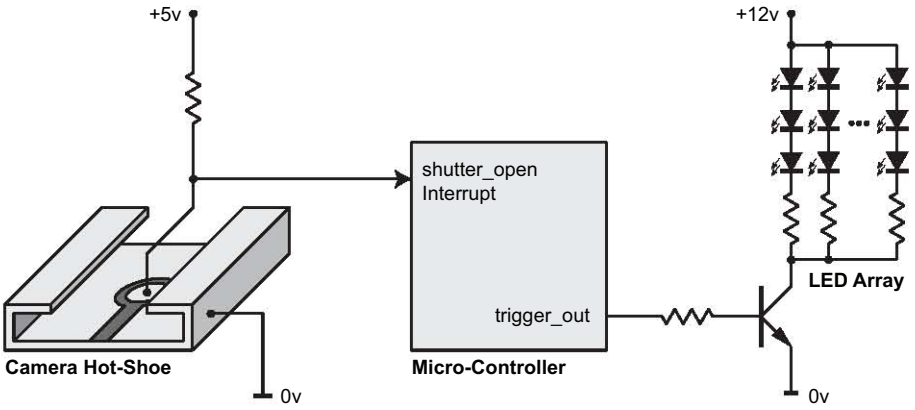



Figure 3.9: Simplified connection diagram of the coded illumination module. One of the micro-controller's external interrupt pins is connected to the camera's hot-shoe terminal in order to detect when the camera shutter is opened. The output of the micro-controller then drives the LED array through a transistor.

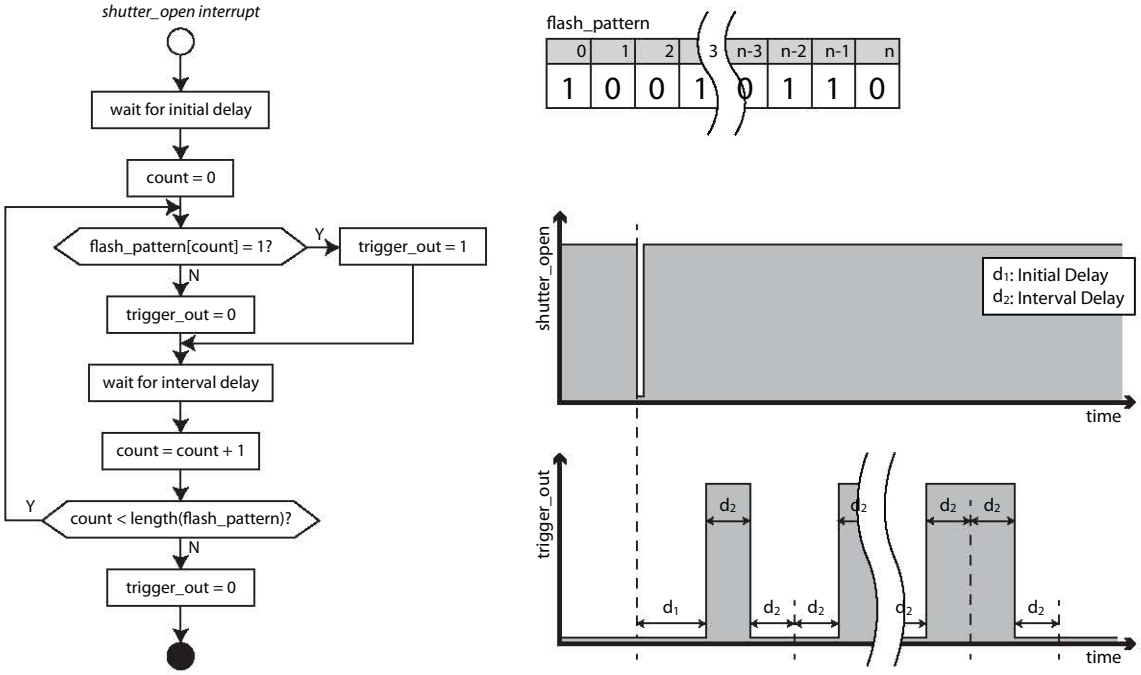



Figure 3.10: Flow-chart and example timing diagrams detailing the functions of the Coded Illumination module. When a new shutter_open interrupt is detected, the micro-controller waits for an initial delay, and then pulses the trigger_out pin according to the pattern stored in the flash_pattern register.

## 3.3 Testing and Performance

### 3.3.1 Coded Aperture

Since the construction of the reconfigurable aperture required disassembling the lens and making structural modifications, there was a chance that the optical elements may have become misaligned, thereby reducing the sharpness of the lens. In order to test for this possibility the performance of the modified lens was compared to that of the unmodified lens.

For a point light source placed in the lens's focal plane, a sharp lens should produce a single point in the image. The image produced from a point source is called the point-spread function (PSF), and any deviation from this ideal would indicate an optical misalignment. A point light source was created by illuminating a small cluster of pixels on an LCD monitor, and this was photographed using the unmodified and modified lenses, both focused at 1.0m. An arbitrary aperture shape was inserted into the modified lens, and the aperture of the unmodified lens was set at f/9.0 since this was calculated to have the same transparent area as the coded shape. Three photographs were taken with each lens at 0.7m (point source in front of the focal plane), 1.0m (point source at the focal plane), and 1.3m (point source behind the focal plane) respectively. The results of this experiment can be seen in figure 3.11. As expected when the point source was placed in front or behind the focal plane the PSFs of the lenses were non-ideal and took the shape of the aperture itself. However when the point source was located on the focal plane both lenses performed equally well, as evidenced by their similar PSFs.

The results of this simple test indicated that the PSF of the focused modified lens was equal to (or at least not significantly worse than) the PSF of the focused unmodified lens. Therefore it was concluded that the modifications made to the lens did not adversely affect the sharpness of the lens. Furthermore, the test quantitatively confirmed that the PSF of an unfocused lens (also known as the blur kernel) takes the shape of the aperture itself.

### 3.3.2 Coded Illumination

The tests that needed to be performed regarding the coded illumination modules included checking the power and coverage of the LED ring array, testing the synchronization timing between the camera and the micro-controller, and finally an overall proof-of-concept test using a coded illumination pattern and a moving target.

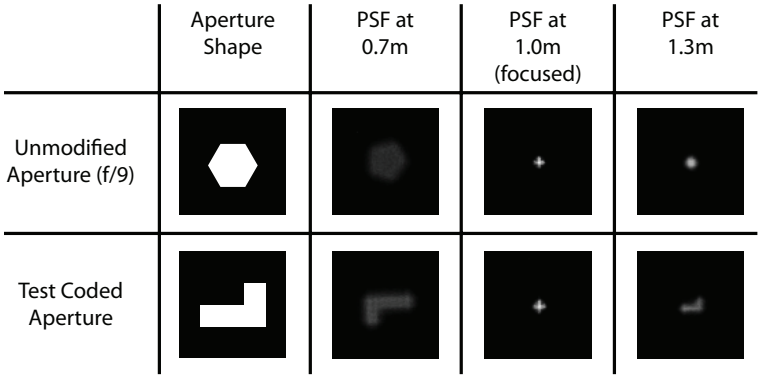



Figure 3.11: A table comparing the PSFs of the lens before and after modification. Both the lenses were focused at 1.0m, and all other camera parameters were held constant.

Since the prototype camera used a curtain shutter it was critical to get the timing during the beginning and end of each exposure correct. Curtain shutters open and close starting from one side of the camera sensor and take a finite amount of time to reach the other side, and therefore not all areas of image sensor are exposed for the same amount of time. While this is not a problem for most photographic applications, it produces noticeable effects when the scene (or in our case, the scene lighting) changes significantly during the exposure. Figure 3.12 shows four photographs of a flat grey card taken with 1/60s exposure times, where the scene was only illuminated near the end of the exposure time. In (a) the scene was illuminated 4.3ms from the end of the exposure time and the entire frame was exposed, however in (b), (c), and (d) the scene was illuminated after the shutter curtain began traversing the sensor, and so the frame was only partially exposed. Therefore in order to prevent the curtain shutter from affecting our results, all coded illumination patterns were timed to complete before 4.3ms from the end of each exposure. Similar tests were performed by illuminating the scene at the beginning of the exposure time, but the entire frame was always properly exposed, implying that the camera only signals an open shutter once the shutter curtain has traversed the entire frame.



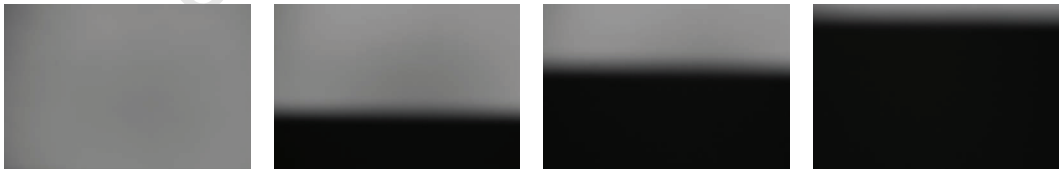



Figure 3.12: Photographs showing the adverse effects encountered when combining time-varying lighting and curtain shutters. In each instance a grey card was illuminated near the end of a 1/60s exposure time, specifically 4.3ms, 3.0ms, 2.3ms and 1.7ms from the end respectively.

In order to test the functionality of the complete system a moving target was photographed with a variety of different illumination patterns. The target consisted of a white horizontal line on a black background moving vertically upward, and it was photographed with a 1/15s exposure time. Figure 3.13 shows a static image of the target in (a), as well as the images produced when using conventional uncoded illumination, strobing (alternating) illumination, and coded illumination in (b), (c), and (d) respectively. The image in (b) depicts the common case of directional motion-blur, while in (c) and (d) the motion-blur has been modulated according to the illumination pattern. The images in figure 3.13 were similar to the expected results, and so the coded illumination module was deemed to be functioning correctly.

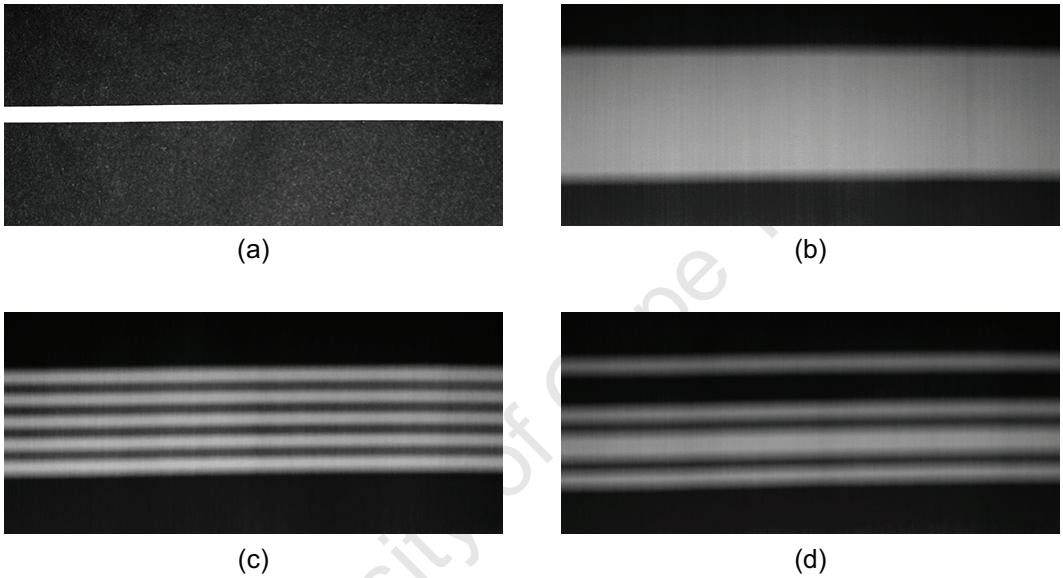



Figure 3.13: Photographs showing the effect of different illumination patterns on a vertically moving line target over a 1/30s exposure time. In (a) the static target is shown, (b) is the result using conventional uncoded illumination, (c) is the result using a strobing (alternating) pattern, and (d) is the result using the arbitrary binary pattern: 101101001.

# Chapter 4

# Coded Aperture Methods

This chapter covers coded aperture theory, and describes how coded apertures can be used for refocussing, depth estimation, and light field applications. The first section explains the function of the aperture in traditional photography by examining geometric camera models, and introducing the concept of the depth-dependent point-spread function (PSF). In the second section, the task of defocus deblurring is dealt with in substantial detail, describing the difficulties associated with traditionally circular apertures, and outlining how a coded aperture can be optimized for deblurring. In a similar way section three covers the task of depth-from-defocus estimation, and explains why optimized coded apertures should outperform circular apertures. Finally in section four, a method for capturing 4D light fields using a coded aperture camera is described, and two popular light field applications are presented, namely, stereo depth estimation and synthesizing virtual photographs.



## 4.1 Theoretical Overview

### 4.1.1 Geometric Camera Models

#### Pin-Hole Camera Model

The ideal pinhole camera forms an inverted image of a scene by only allowing light rays passing through an infinitesimally small aperture to reach the image plane. As seen in figure 4.1, only one light ray originating from each point in the scene can pass through the aperture, and while this severely limits the total amount of light entering the camera, it produces sharp images of every object in the scene regardless of their distance. Ideal pinhole cameras are impossible to construct physically, and near-ideal pinhole cameras let in so little light that the required exposure times become impractically long [10].

**Thin-Lens Camera Model**

The infinitesimally small aperture of the pinhole camera can be replaced with a lens in order to collect more light and thereby reduce the exposure time required to capture an image. However, only light rays originating from points at a specific distance from the lens (called the *focal plane*) are refracted in such a way as to converge to a single point on the image sensor. The rays originating from all other points either get bent too much or too little, and so form circular shapes (known as *circles of confusion*) on the image plane. The superposition of many of these circles of confusion produces the effect known as out-of-focus blur. Figure 4.2 shows how rays from a point on Object 2 (which is located on the focal plane) converge to a point on the image plane, while rays from a point on Object 1 are bent too much and form a circle of confusion on the image plane [10].

A compromise between the sharpness of the pinhole camera and the brightness of the thin-lens camera can be achieved by adding an aperture to the thin-lens model. The aperture restricts the number and spread of light rays passing through the lens to the image plane, and therefore the smaller the aperture, the closer the model resembles the pinhole model. Most cameras have an adjustable aperture which allows the photographer to choose between a sharper image or a brighter image depending on the environmental conditions. Figure 4.3 shows how the added aperture reduces the size of the circle of confusion when compared to figure 4.2. Lastly, it is important to note that the shape of an out-of-focus point on the image plane is not always circular, but is instead determined by the shape of the aperture itself [39].

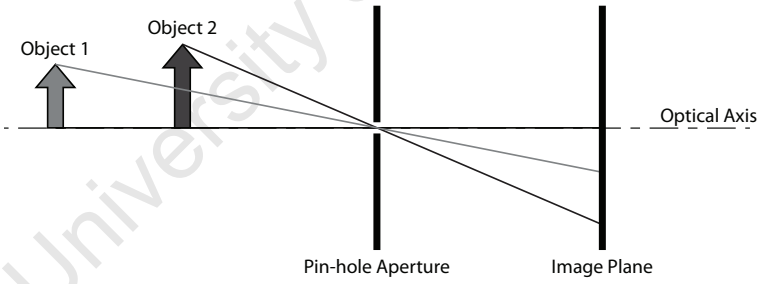



Figure 4.1: Diagram showing the geometric model of an ideal pinhole camera imaging two objects at different distances from the aperture. The infinitesimally small aperture creates sharp images of both objects.

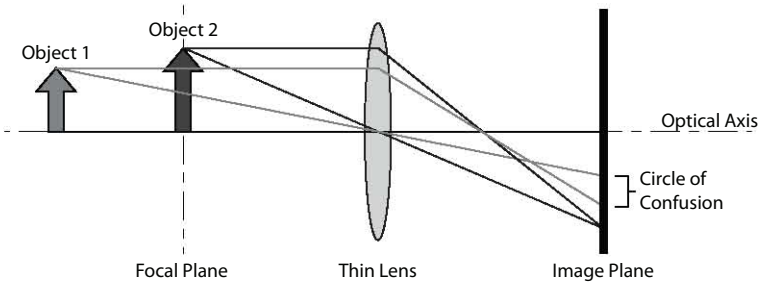



Figure 4.2: Diagram showing the geometric model of a simplified thin-lens camera imaging two objects at different distances from the lens. The large lens area gathers much more light than a pinhole camera, but only objects located in the focal plane produce sharp images.

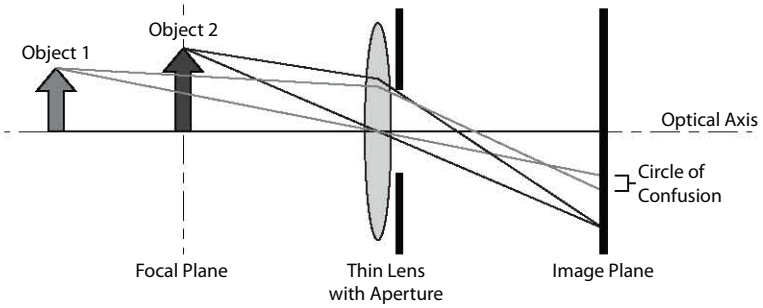



Figure 4.3: Diagram showing the geometric model of a simplified thin-lens camera with a finite sized aperture. While the aperture restricts the amount of light entering the camera it also reduces the level of blurring for objects that lie outside the focal plane.

### 4.1.2 The Point-Spread Function as a Linear Filter

The point spread function (PSF) is defined as the image produced by a camera of an ideal point source, and is analogous to the impulse response of a generalized 2D linear system. For example, the PSF of the ideal pinhole camera model described in the previous section is simply a point, since points in the scene are mapped perfectly to points on the image plane. When considering lens-based cameras, the PSF becomes a function of the distance between a scene point the focal plane, since the PSF for points in the focal plane is an ideal point, but as the point moves away from the focal plane the PSF grows in size and takes a shape determined by the shape of the aperture. Therefore each position in a photograph may have a unique PSF, determined by the depth of the scene object imaged at that location [18].

Since the PSF describes the image formed of a point source, and since all scenes can be decomposed into an infinite collection of point sources, the image of an entire planar scene can be calculated using a single PSF and the principle of linear superposition. Therefore the image formed of a planar scene can be thought of as the two-dimensional convolution of the scene's ideal sharp image and the appropriate PSF for that depth.

Figure 4.4 shows a simplified model of a thin-lens camera (with a circular aperture) imaging three planar objects, each at a different distance from the lens. Object 2 lies on the focal plane, while objects 1 and 3 lie in front of and behind the focal plane respectively. The same situation is described again in figure 4.5, but this time as a linear combination of three separate convolution operations between the ideal sharp images of the scene objects and their depth-varying PSFs. As expected, the PSF for Object 2 is a single point, while the PSFs for Objects 1 and 3 are circles with significant area. The convolution between any input and a single point is equal to the input itself, and so the representation of Object 2 in the final image is identical to its ideal sharp image. In contrast, the convolutions of the other two objects with their circular PSFs create significantly blurred representations in the final image.

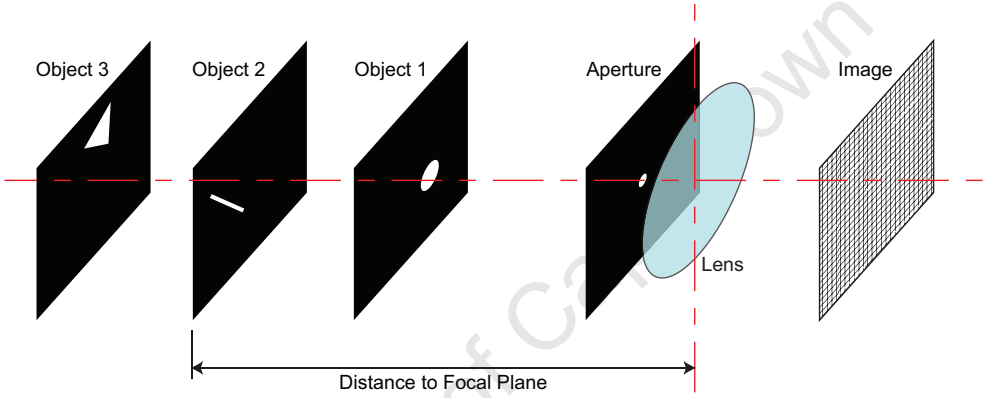



Figure 4.4: Diagram showing a simplified model of a thin-lens camera imaging three planar objects, each at a different distance from the lens. The black areas in each plane are perfectly transparent and the white areas are completely opaque.

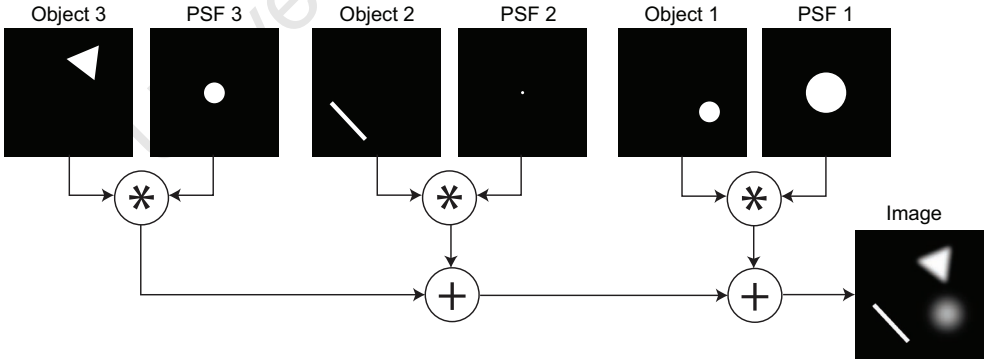



Figure 4.5: A block-diagram describing the same situation as in figure 4.4, but now as a linear combination of three separate convolution operations between the ideal sharp images of the scene objects and their depth-varying PSFs.

## 4.2 Refocusing with Coded Apertures

### 4.2.1 Overview

As discussed in section 4.1 above, out-of-focus blur produced by a thin-lens camera can be modeled as the convolution of an ideal sharp image with a PSF with significant non-zero area. This can be written as

$$y = x * f_k, \tag{4.1}$$

where $x$ is the ideal sharp image, $f_k$ is the depth-dependent PSF blur kernel, and $y$ is the observed blurry image. Since convolution in the spatial domain is equivalent to multiplication in the frequency domain, the process can also be described as

$$Y = X \cdot F_k, \tag{4.2}$$

where $X$, $F_k$, and $Y$ are the Fourier transform pairs of $x$, $f_k$, and $y$ respectively. Figure 4.6 graphically depicts the above relationships, and shows the characteristic *sinc* shape of the circular PSF's frequency spectrum (for simplicity only a one-dimensional slice of the full two-dimensional spectrum is shown). This sinc shape acts as low-pass filter by attenuating the high-frequency components in the sharp image, and also completely removes frequency components located at any of its regularly spaced zero values. This explains why out-of-focus images lack high frequency elements such as hard edges and small-scale texture [39].

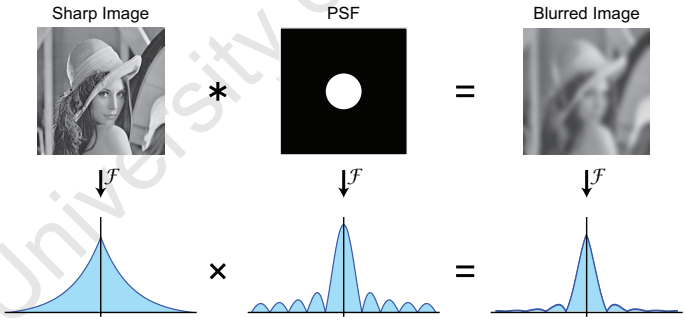



Figure 4.6: Diagram describing out-of-focus blur in both the spatial and frequency domains (where $\mathcal{F}$ represents the Fourier Transform). The PSF acts as a low-pass filter by attenuating high-frequency components.

Refocusing (or deblurring) is the process of attempting to reconstruct the true image, $x$, from the observed blurry image $y$. Assuming that knowledge of the blur kernel is provided, a naïve refocusing method could be obtained by rearranging equation 4.2 as

$$X = Y/F_k. \tag{4.3}$$

This suggests that the sharp image can be obtained by simply dividing the Fourier transform of the blurred image by the Fourier transform of the PSF. Unfortunately this relationship is badly defined since it places the zero and near-zero values of $F_k$ in the denominator. Another way to look at it is that during the initial blurring process multiplication by zeros has irreversibly removed some information from the image, and therefore true refocusing becomes impossible. Even where the magnitude of $F_k$ is only near zero, the original information becomes lost amidst the background noise, and dividing by these small values serves only to amplify the noise. Despite these difficulties, out-of-focus deblurring has remained a popular area of research and many implementations have been developed that significantly reduce the amount of blur in an image.

Fortunately not all aperture shapes produce PSFs with absolute zero values in their frequency spectra [39]. For example, figure 4.7 compares the spectra of a regularly shaped rectangular blur kernel (the one-dimensional equivalent to a circular two-dimensional PSF) with that of a far more irregularly shaped kernel. As expected the rectangular kernel has a sinc-shaped spectrum with regularly shaped zero values, but in contrast the irregular kernel's minima are irregularly spaced and never quite reach absolute zero. Therefore, while using such an aperture shape would still attenuate high-frequency information, no frequency band would be lost completely, and this would certainly improve the feasibility of equation 4.3. It should also be noted that the irregular shape in figure 4.7 was randomly selected and is not meant to represent the optimal pattern. Finding the optimal aperture shape is dealt with in detail in the next section.

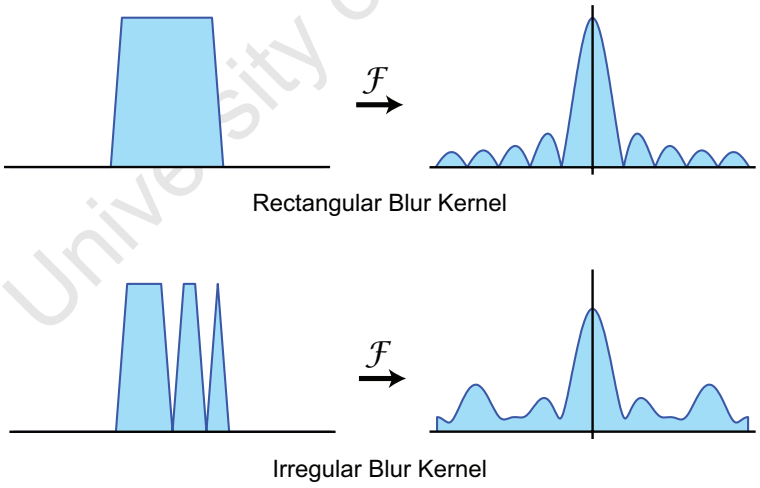



Figure 4.7: A diagram comparing a rectangular and an irregular blur kernel in the frequency domain. The rectangular kernel has regularly spaced zero values in its magnitude spectrum, while the irregular kernel does not.

### 4.2.2 Selecting the Aperture Pattern

#### Aperture Shape Representation

Selecting the optimal aperture shape involves devising a performance metric and then searching through the solution-space evaluating and comparing the possible shapes. However, before either of these two tasks can be performed, a discrete mathematical representation of an aperture shape needs to be defined. Since all the coded apertures used in our experiments were formed out of $N \times N$ binary grids, the obvious representation is that of an $N \times N$ binary array. Each element (logical bit) in the array represents whether the corresponding block in the physical aperture is opaque (if it is a 0) or transparent (if it is a 1), and takes the form

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N-1} & a_{1,N} \\ a_{2,1} & a_{2,2} & \dots & a_{2,N-1} & a_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{N-1,1} & a_{N-1,2} & \dots & a_{N-1,N-1} & a_{N-1,N} \\ a_{N,1} & a_{N,2} & \dots & a_{N,N-1} & a_{N,N} \end{bmatrix}. \tag{4.4}$$

In some cases a vector (or bit string) representation of an aperture is preferred over the array representation defined above. In these cases the original array is merely vectorized by concatenating each row, one after the other in the following manner:

$$\mathbf{a} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,N} & a_{2,1} & a_{2,2} & \dots & a_{2,N} & a_{N,1} & a_{N,2} & \dots & a_{N,N} \end{bmatrix}. \tag{4.5}$$

Where a formal definition of the relationship between $A$ and $\mathbf{a}$ is required, the following can be used:

$$\mathbf{a} = \sum_{i,j} P_{ij} \cdot a_{i \cdot N + j}, \tag{4.6}$$

where $a_{ij}$ is the element in the $i^{th}$ column of the $j^{th}$ row in $A$, and $P_{ij}$ is a matrix defined as

$$P_{ij}(x, y) = \begin{cases} 1, & \text{for } [x, y] = [i, j] \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

The primary use of the bit-string vector representation, $\mathbf{a}$, is that it can be converted to a decimal number, allowing an $N \times N$ aperture to be uniquely defined by a value between $0$ and $2^{N^2} - 1$.

**Aperture Performance Metric**

Equation 4.3 states that the process of defocus deblurring requires dividing the Fourier transform of the observed blurry image by the Fourier transform of the PSF for that particular depth. It has been discussed how zero or near-zero values in the PSF's frequency spectrum make this problem ill-defined and magnify noise. Therefore, it could be intuitively hypothesized that apertures with larger minimum magnitudes in their frequency spectra will perform better than apertures with lower minima. This is a simple performance criterion and the optimal aperture shape would be the one that minimizes the fitness function

$$R(F_k) = 1/min(|F_k|). \tag{4.8}$$

However, this fitness function makes some simplistic assumptions (e.g. all frequency components in the sharp image are equally important), and completely ignores the effects of non-minimum values in the frequency spectra. Therefore, a more rigorous performance metric was sought, which ideally would be derived analytically from statistical models of natural images. The fitness function that was finally selected was developed by Zhou and Nayar [39]. A summarized derivation is presented here, but their paper should be consulted for a more in-depth explanation.

Taking noise into consideration, equations 4.2 and 4.3 (which describe the formation of defocus blur) can be extended to

$$y = x * f_k + n \tag{4.9}$$

and

$$Y = X \cdot F_k + \zeta, \tag{4.10}$$

where $n$ is additive Gaussian white noise with a distribution of $N(0, \sigma^2)$, and $\zeta$ is its two-dimensional Fourier transform. The derivation starts with the fundamental assumption that a high-performing aperture will result in a good estimate of the sharp image, or in other words, that the performance of an aperture is inversely determined by the error between the estimated sharp image and the true sharp image. Therefore the fitness function can be defined as:

$$R(F_k, X) = \mathop{\mathbb{E}}_{\zeta}[\|\hat{X} - X\|^2], \tag{4.11}$$

where $R$ is the fitness function (with a smaller value indicating a better performance), and is calculated as the expectation (with respect to noise) of the $L_2$ distance between the estimated sharp image $\hat{X}$ and the true image $X$. The quantity $\hat{X}$ itself is the solution to the maximum *a posteriori* (MAP) problem

$$\hat{X} = \arg\max_X P(X|Y, F_k) = \arg\max_X P(Y|\hat{X}, F_k) \cdot P(\hat{X}), \tag{4.12}$$

which states that the estimated sharp image is the one that is most likely, given the observed image $Y$ and the PSF kernel $F_k$. Assuming a Gaussian model and by calculating the logarithmic energy function, the MAP equation above can be solved as the minimization of

$$E(\hat{X}|Y, F_k) = \|\hat{X} \cdot F_k - Y\|^2 + H(\hat{X}), \tag{4.13}$$

where $H(\hat{X})$ is a regularization term based on image priors. The $\hat{X} \cdot F_k$ term takes the estimated sharp image and re-blurs it with the PSF, and so the $L_2$ distance between it and the original blurred image $Y$ can then be thought of as the error in the estimation. By defining the regularization term as $H(\hat{X}) = \|C \cdot \hat{X}\|^2$ and minimizing equation 4.13, the popular Wiener deconvolution is derived:

$$\hat{X} = \frac{Y \cdot \bar{F}_k}{|F_k|^2 + |C|^2}, \tag{4.14}$$

where $\bar{F}_k$ is the complex conjugate of $F_k$, and the optimal $|C|^2$ is known to be a matrix of the noise-to-signal ratios (NSR), namely $|\frac{\sigma}{X}|^2$. Wiener deconvolution is simply an implementation of frequency domain deconvolution that attempts to minimize the level of noise at frequencies with low signal-to-noise (SNR) ratios. Unfortunately $X$ is not usually known, and so $C$ cannot be calculated exactly. Therefore $C$ is often estimated by a scalar or a matrix based on derivative filters, but in our case it will be optimized near the end of the derivation. Substituting equation 4.14 into equation 4.11 gives:

$$R(F_k, X, C) = \mathop{\mathbb{E}}_{\zeta} \left\| \frac{\zeta \cdot \bar{F}_k - X \cdot |C|^2}{|F_k|^2 + |C|^2} \right\|^2. \tag{4.15}$$

Since $\zeta$ is assumed to be Gaussian white noise with a distribution of $N(0, \sigma^2)$, this can be simplified to

$$R(F_k, X, C) = \left\| \frac{\sigma \cdot \bar{F}_k}{|F_k|^2 + |C|^2} \right\|^2 + \left\| \frac{X \cdot |C|^2}{|F_k|^2 + |C|^2} \right\|^2. \tag{4.16}$$

The above definition of the aperture fitness function is now dependent on the PSF, the true sharp image, and the NSR. However, since $X$ is a sample from the space of all natural images and the fitness function must be valid for all natural images, we can remove $X$ from the list of parameters by calculating the expectation over all possible $X$ values:

$$R(F_k, C) = \mathop{\mathbb{E}}_{X}[R(F_k, X, C)] = \int_X R(F_k, X, C) \cdot d\mu(X), \tag{4.17}$$

where $\mu(X)$ is the measure of sample $X$ in the entire image space. Furthermore, following the $1/f$ law for natural images, we know the expectation of $|X|^2$ exists. If we define this

expectation as

$$A(\omega) = \int_X |X(\omega)|^2 d\mu(X), \tag{4.18}$$

where $\omega$ is the two-dimensional frequency variable, then we can simplify the fitness function to

$$R(F_k, C) = \left\| \frac{\sigma \cdot \bar{F_k}}{|F_k|^2 + |C|^2} \right\|^2 + \left\| \frac{A^{1/2} \cdot |C|^2}{|F_k|^2 + |C|^2} \right\|^2. \tag{4.19}$$

Minimizing $R(C|K)$ for any value of $F_k$ gives

$$|C|^2 = \frac{\sigma^2}{A}, \tag{4.20}$$

and substituting this result into equation 4.19 finally gives the following metric that can be used to evaluate the performance of an aperture shape [39]:

$$R(F_k) = \sum_{\omega} \frac{\sigma^2}{|F_k(\omega)|^2 + \frac{\sigma^2}{A(\omega)}}. \tag{4.21}$$

For each frequency $\omega$, the above metric indicates the degree to which that noise at that frequency will be amplified. Therefore, the PSF that minimizes $R(F_k)$ will be the PSF that amplifies noise the least across all frequencies, thereby producing the best deblurring results. The value of $A$ can be set theoretically by enforcing the $1/f$ law, or it can be practically calculated by averaging the power spectra of many natural images. It is important to remember that the PSF that minimizes $R(F_k)$ is optimal for the entire space of natural images, and might not be optimal for any specific image. It is also interesting to note the strong dependence on the level of noise present in the original image, and this suggests that different aperture shapes will be optimal for different levels of noise [39].



**Searching for the Optimum Aperture**

As previously mentioned, for a square binary matrix of size $N \times N$, there are $2^{N \times N}$ possible combinations. For example, values for $N$ of 3, 7 and 13 result in 512, $5.63 \times 10^{14}$, and $7.48 \times 10^{50}$ unique combinations respectively. Clearly for values of $N$ greater than 5 or 6, the sheer size of the solution space makes an exhaustive search for the optimally performing aperture infeasible. Since we intend to test apertures with $N$ values ranging from 7 to 13, we require a more intelligent search method. The method that was chosen was suggested by Zhou and Nayar [39], and is a simple variant of the popular genetic algorithm optimization method.

The algorithm is described graphically as a flow-chart in figure 4.8. It requires six parameters, namely: $G$ the maximum number of generations allowed, $S$ the total number of vectors

in the initial population, $L$ the length of each vector (which is alway equal to $N \times N$), $M$ the number of best performing vectors that survive each generation, $c1$ the probability of crossover, and finally $c2$ the probability of mutation. Each binary vector in the population is of the form described by equation 4.5 and represents a single aperture shape instance.

In order to initialize the population, the algorithm begins by randomly generating $S$ binary vectors of length $L$. Each vector in the population is then rated according to the chosen fitness function, and the best $M$ performers are kept as a breeding pool while the rest of the population is discarded. The population (now of size $M$) is then repopulated through a process of crossover and mutation, using parent vectors from the breeding pool. The crossover component creates children by swapping the bits between two parents with probability $c1$, and the mutation component inverts the bits in the children with probability $c2$. The combination of crossover and mutation provides a balance between learning from past successful vectors and exploring the rest of the solution space, and therefore the values $c1$ and $c2$ must be tuned for optimal performance. Once the population has been repopulated to its original size $S$, the process repeats itself until the maximum number of generations $G$ has been reached.

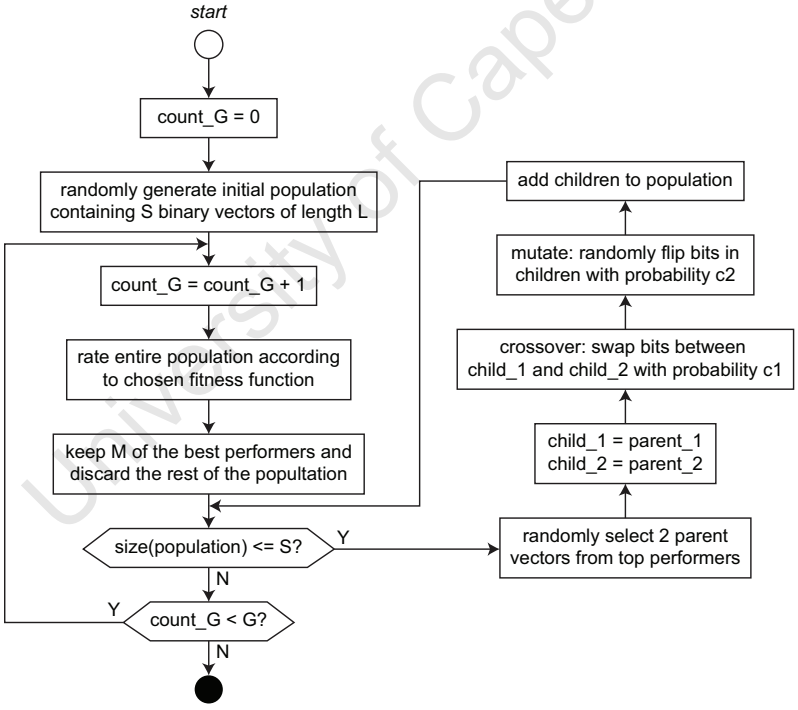



Figure 4.8: A flow-diagram describing the genetic algorithm used to search for the optimum aperture shape.

### 4.2.3 Deblurring

Once an optimal aperture shape has been selected, and it has been used to capture an image with defocus blur, a deblurring (or deconvolution) algorithm must be used to estimate the original sharp image. While the general process of deblurring and the problems associated with it have been discussed earlier, in this section we select a specific algorithm and explain its individual characteristics.

Since it has been suggested that using an optimized coded aperture instead of a conventional aperture fundamentally improves the definition of the deblurring problem, it should improve the results obtained regardless of which specific algorithm is used. Therefore the choice of algorithm is largely non-vital, and was therefore based primarily on practical considerations such as ease of implementation, efficiency, and visual performance. However it is still important to formally define the choice of de-blurring implementation (as well as its associated parameters) for the sake of reproducibility, and in order to accurately compare the performance of different apertures.

The implementation that was chosen for our experiments was developed by Levin et al. [18]. Their method uses a statistical image prior based on the sparse derivative distributions found in natural images. This prior prefers the output of the algorithm to have smooth gradients in the majority of the image, and large gradients (such as object edges) in only a few specific locations. A basic overview of the algorithm is be presented below, but for a more in-depth analysis see the appropriate section in the specified paper.

The algorithm starts by defining a model that gives the probability that an image $x$ is the correct sharp representation for the observed blurry image $y$:

$$P_k(x|y) \propto e^{-(\frac{1}{\eta^2}|C_{f_k}x-y|^2+\alpha|C_{g_x}x|^2+\alpha|C_{g_y}x|^2)}, \tag{4.22}$$

where $\eta^2$ is the average noise power, $\alpha$ is a tunable scalar, $g_x$ and $g_y$ are the horizontal and vertical derivative filters, and $C_{f_k}$, $C_{g_x}$, and $C_{g_y}$ are the convolution matrices for filters $f_k$, $g_x$, and $g_y$ respectively. $C_{f_k}x$ is the estimated sharp image re-blurred with the PSF, and so the $|C_{f_k}x - y|^2$ term is an $L_2$ measure of the error in the estimation. In essence this model gives a high probability to an image $x$ if it, when blurred, is visually similar to the observed image $y$, and if it has smooth gradients in most of the image. The $x$ that maximizes this probability function becomes the most likely estimate of the sharp image:

$$\hat{x} = \arg\max_x P_k(x|y) = \arg\min_x (\frac{1}{\eta^2}|C_{f_k}x - y|^2 + \alpha|C_{g_x}x|^2 + \alpha|C_{g_y}x|^2). \tag{4.23}$$

Maximizing equation 4.22 requires minimizing the bracketed term in the exponent of $e$, and when a Gaussian distribution is assumed for the gradient functions the optimal solution can

be calculated using a sparse set of linear equations in the form $A\mathbf{x} = \mathbf{b}$, where:

$$A = \frac{1}{\eta^2}(C_{f_k})^T C_{f_k} + \alpha(C_{g_x})^T C_{g_x} + \alpha(C_{g_y})^T C_{g_y} \quad \text{and} \quad \mathbf{b} = \frac{1}{\eta^2}(C_{f_k})^T y. \tag{4.24}$$

The above optimization can be solved efficiently in the frequency domain, but since natural image do not actually follow a Gaussian derivative distribution, using this assumption tends to over-smooth the output image. This smoothing can be avoided by assuming that the derivative distributions follow a more heavy-tailed function such as $\rho(z) = |z|^\beta$, but by using this non-Gaussian assumption the minimization of equation 4.23 then becomes

$$\hat{x} = \arg\min_x \left( |C_{f_k} x - y| + \sum_{i,j} [\rho(x(i,j) - x(i+1,j)) + \rho(x(i,j) - x(i,j+1))] \right), \tag{4.25}$$

which cannot be solved with linear equations, nor in any other closed form [18]. In our experiments the Gaussian assumption was used when deblurring as in intermediate step, and the heavy-tailed sparse assumption was only used when calculating final results. This gives a good compromise between the processing efficiency of the Gaussian assumption and the visual performance of the sparse assumption.

## 4.3 Estimating Depth with Coded Apertures

### 4.3.1 Overview

The task of obtaining a depth estimate from an out-of-focus image is conceptually similar to the task of deblurring, which was discussed in the previous section. However, instead of attempting to reverse the blurring process, the specifics of the blur are analyzed in order to estimate the underlying scene geometry.

As discussed in section 4.1, a non-ideal PSF is produced when light rays originating from an object outside the focal plane fail to converge to a single point on the image plane. Instead the rays are either bent too much or too little, and thereby illuminate a finite-sized region on the image plane called the circle of confusion. The shape of the PSF is equal to the shape of the aperture itself, and its scale is proportional to the distance between the focal plane and the object being imaged. This means that if the scale of the PSF and the location of the focal plane are known, then the distance of the object can be deduced [18]. Figure 4.9 shows an example of how the scale of a PSF changes proportionally with the distance from the focal plane. The figure also shows the subtle shape inversion that occurs when passing from one side of the focal plane to the other, and how it is only visible when using a non-symmetric aperture.

Therefore estimating depth-from-defocus is primarily a problem of determining the PSF that caused the blur in the image. This is a non-trivial problem since the true sharp image is not usually known, so the PSF cannot be obtained by simply rearranging equation 4.3 as was the case for deblurring. However, the general shape of the PSF is already known, since it is determined by the shape of the physical aperture in the camera, and so only its scale must be determined. Naïvely one might attempt to find the PSF scale in the spacial domain by searching for shapes in the observed image that match scaled versions of the known PSF shape. However, as shown in figure 4.10, this is completely infeasible for all but the most trivial cases: the image shows that while the PSF shape can be seen in (a), (b), and (c), which only include one or two point source targets, the task becomes much more difficult as the number of point sources increases (tending towards a continuous scene of infinite point sources). This problem is complicated further by the fact that the overlapping of PSFs during convolution occurs non-linearly when the minimum and maximum limits of the image sensor are reached (in under- or overexposed areas in the image).

A far more effective method for determining the PSF scale is by analyzing the frequency spectrum of the observed image. As described in section 4.2 and shown in figure 4.6, the frequency spectrum of the observed blurry image can be modeled by the multiplication of the sharp image and PSF frequency spectra. Since anything multiplied by zero is zero, the

zeros in the PSF's spectrum are also present in the observed image's spectrum, regardless of the shape of the sharp image's spectrum. We also know from Fourier theory that an increase in the spacial domain scale causes a decrease in the frequency domain scale and vice versa. Therefore, as shown in figure 4.11, a shift in PSF scale significantly alters the zero locations in its spectrum. This fact, coupled with the relative ease with which zeros in the observed image can be located, makes it a robust method for estimating the PSF scale [18].

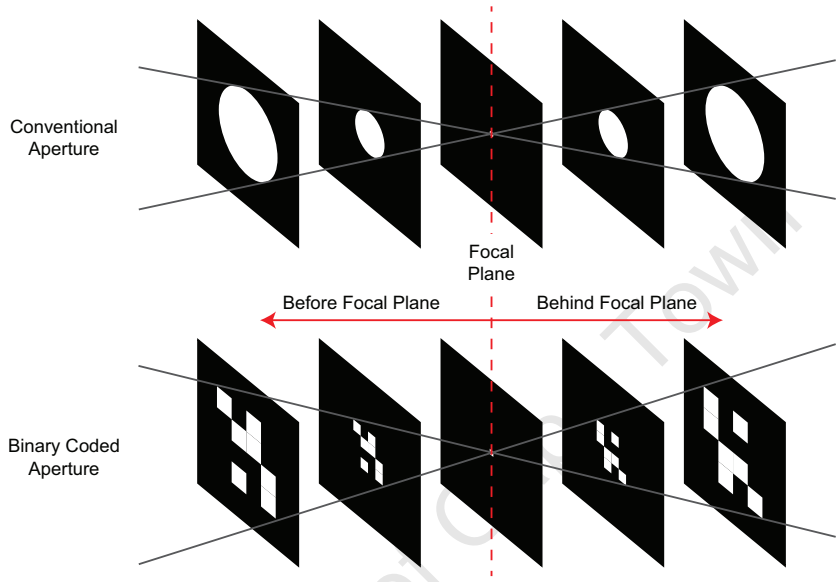



Figure 4.9: Diagram describing how the scale of a PSF varies with the distance between the focal plane and the target object. The PSF is actually inverted for objects located before the focal plane, but this is only visible when using a non-symmetric aperture.

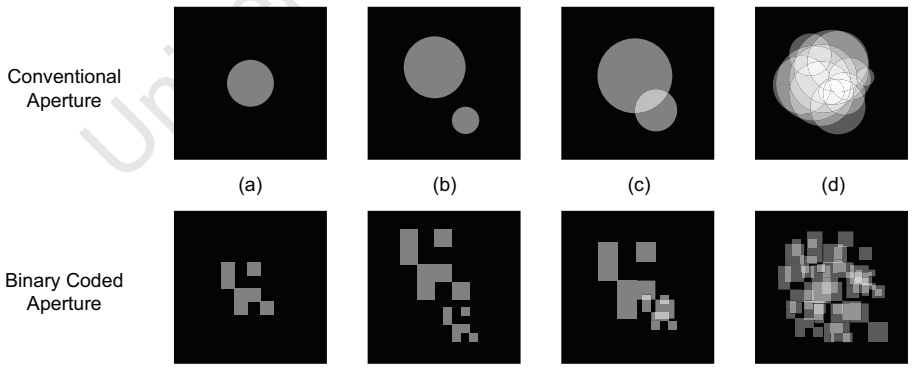



Figure 4.10: Diagram showing the challenge of identifying blur scale for conventional and coded apertures in the spatial domain. Each aperture is shown in four different situations: (a) a single point source, (b) two non-overlapping point sources, (c) two overlapping point sources, and (d) multiple overlapping point sources tending towards a continuous scene.

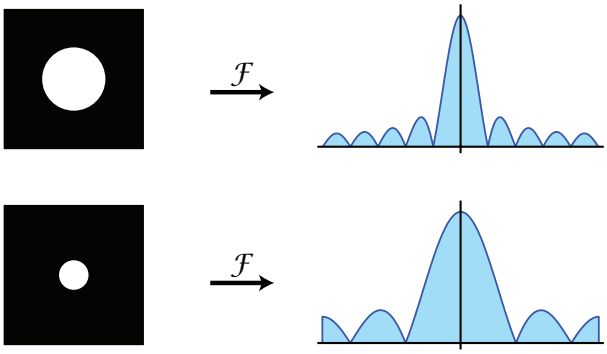



Figure 4.11: Diagram showing the effect of scaling a PSF in the frequency domain. Following the properties of the Fourier transform, a decrease in scale in the spatial domain is mirrored by an increase in scale in the frequency domain. Note the evident change in zero locations.

Determining blur scale using zero locations does include some specific challenges. Firstly, additive noise is always present in the observed image. This prevents the values in its frequency spectrum from ever truly reaching zero, and therefore the task of locating the zeros is made more complicated. Secondly, zeros in the observed image spectrum may have originated from a zero in the sharp image's spectrum rather than from that of the PSF. Therefore not all zeros in the observed image are related to PSF scale. Lastly, the zeros within the spectrum of a conventional round aperture are regularly spaced, and therefore distinct PSF scales may have coincident zero locations, increasing the chances of misidentification [18].

### 4.3.2 Selecting the Optimal Aperture Shape

#### Aperture Shape Representation

The digital representations used to define aperture shape in this section are identical to the ones described for the task of deblurring (section 4.2). The array and vector representations are defined in equations 4.4 and 4.5, and therefore they receive no further explanation in this section.

#### Aperture Performance Metric

Before the optimal aperture shape for estimating depth can be found, a performance metric must be devised that can be used to compare the performance of different apertures. As mentioned previously, obtaining a depth estimate is achieved by determining the scale of the PSF that caused the blurring in the observed image, and the most robust method for determining the blur scale is to locate the zero locations in the observed image's frequency spectrum. However, to achieve accurate results the zero positions for each PSF scale should be unique, or else the chance of misidentification is increased. The frequency spectrum of

a conventional round aperture is in the shape of a two-dimensional sinc function, and this shape has zero positions spaced regularly over the frequency domain. Therefore different PSF scales tend to have coincident zero locations. Fortunately this is not true of all aperture shapes, and in the case of binary coded apertures the zero locations can actually be engineered into an optimal configuration.

Figure 4.12 shows three scales of an arbitrary one dimensional coded aperture, together with their Fourier transforms. In the figure, scale 1 has a pair of zeros located at frequency $\omega_1$, and scale 2 has a pair of zeros located at frequency $\omega_2$. Intuitively it can be supposed that reliable scale identification requires that at frequency where one scale has a zero, the other scales should have significant content. For example, if a zero was found at frequency $\omega_2$ in an observed image, we can be quite confident that the PSF was at scale 2, since both of the other scales have significant content at $\omega_2$.

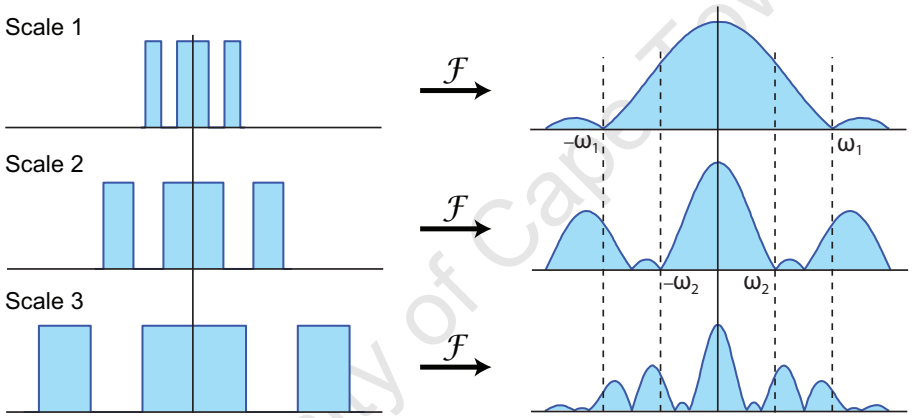



Figure 4.12: Diagram showing the zero locations in the frequency domain for three different scales of a one-dimensional binary aperture. Good scale discrimination requires that at a frequency where one scale has a zero the other scales should have significant content, and vice versa. Diagram taken from [18].



However logical it might seem, there is no reason to believe that the intuitive criterion presented above is the optimal way to rate the depth performance of a particular aperture. Instead, we prefer to use an analytically derived criterion, based on strong statistical information and the mathematical definition of the depth estimation problem itself. The criterion that was chosen is developed by Levin et al. [18], and a brief overview of its derivation is presented below. For a more in depth explanation see the referred paper.

The derivation begins by defining the sparse derivative distribution prior for natural images, and assuming that the distribution is Gaussian with a zero mean:

$$P(x) \propto \prod_{i,j} e^{-\frac{1}{2}\alpha\left((x(i,j)-x(i+1))^2+(x(i,j)-x(i,j+1))^2\right)} = N(0,\Psi) \tag{4.26}$$

and

$$\Psi^{-1} = \alpha((C_{g_x})^T C_{g_x} + (C_{g_y})^T C_{g_y}), \tag{4.27}$$

where $(i,j)$ are the pixel indices, $\alpha$ is a scalar set so that the variance of the prior distribution matches that of natural images, $N$ is the normal distribution, $g_x$ and $g_y$ are the horizontal and vertical derivative filters, and $C_{g_x}$ and $C_{g_y}$ are the convolution matrices for $g_x$ and $g_y$ respectively. The function $P(x)$ therefore gives a high probability for an image $x$ that follows the sparse derivative prior and consists mostly of smooth gradients, with only a few pixels (such as object boundaries) having significant derivative values. The Gaussian assumption is not as accurate as assuming that the distribution is heavy-tailed, but its use is justified because it greatly simplifies the mathematics involved while still producing acceptable results. The same prior can also be expressed in the frequency domain as

$$P(X) \propto e^{-\frac{1}{2}\alpha X^T \bar{\Psi}^{-1} X}, \tag{4.28}$$

where $\bar{\Psi}^{-1}$ is the Fourier transform of $\Psi^{-1}$ and takes the form of

$$\bar{\Psi}^{-1} = \alpha \cdot diag(|G_x(v,\omega)|^2 + |G_y(v,\omega)|^2) \tag{4.29}$$

where $(v,\omega)$ are the coordinates in the frequency domain, and $G_x$ and $G_y$ are the Fourier transforms of $g_x$ and $g_y$. Since the observed image $y$ is produced by convolving the sharp image $x$ and the PSF $f_k$ (equation 4.9), the probability distribution $P_k(y)$ of observed images blurred with the PSF $f_k$ can be calculated as a linear transform of the image prior for sharp images $P(x)$. $P_k(y)$ will also then be a Gaussian distribution, but with its variance transformed by $f_k$:

$$P_k(y) \sim N(0,\Sigma_k) \tag{4.30}$$

where the covariance matrix $\Sigma_k$ is a linearly transformed version of the prior covariance with added noise

$$\Sigma_k = C_{f_k}\Psi(C_{f_k})^T + \eta^2 I \quad \overset{\mathcal{F}}{\Rightarrow} \quad \bar{\Sigma}_k = C_{F_k}\bar{\Psi}(C_{F_k})^T + \eta^2 I, \tag{4.31}$$

where $\bar{\Sigma}_k$ and $C_{F_k}$ are the Fourier transforms of $\Sigma_k$ and $C_{f_k}$ respectively.

However, since the Fourier transform of a convolution matrix is always a diagonal matrix [18], $P_k(y)$ can be simplified to

$$P_k(y) \propto exp\left(\frac{-\frac{1}{2}\sum_{v,\omega}|Y(v,\omega)|^2}{\sigma(v,\omega)}\right), \tag{4.32}$$

where $\sigma(v,\omega)$ are the diagonal entries of $\bar{\Sigma}_k$:

$$\sigma(v,\omega) = \frac{|F_k(v,\omega)|^2}{\alpha|G_x(v,\omega)|^2 + \alpha|G_y(v,\omega)|^2} + \eta^2. \tag{4.33}$$

The function $P_k(y)$ stated in equation 4.32 can now be used to calculate the probability that an observed image $Y$ was blurred with a particular PSF scale $F_k$. Unsurprisingly the equation also functions as a formalized version of the intuitive method of locating zero positions in the frequency spectrum of the observed image. For positions where $F_k$ has zero values in its spectrum but $Y$ does not, $|Y(v,\omega)|^2/\sigma(v,\omega)$ becomes strongly positive. The summation of many such occurrences (such as in an observed image that was not blurred by $F_k$) will tend to drive the value of $P_k(Y)$ low, indicating a low probability that $Y$ was blurred with $F_k$. Furthermore, the presence of the derivative filters $G_x$ and $G_y$ in the denominator of equation 4.33 indicates that the criterion gives more weight to zeros in $F_k$ that are located at low frequency positions, since that is statistically where natural images have the majority of their content.

Now that a model determining the probability that an observed image $y$ was blurred with PSF $f_k$ has been defined, we can use it to develop a performance metric for a particular PSF. The different scales of a well performing PSF shape should be easily and reliably distinguished, while the scales of a poor PSF shape will tend to be ambiguous and hard to detect. Since two scales, $k_1$ and $k_2$, of a particular PSF $f_k$ that are easily distinguishable will produce different $P_k$ distributions, we can use the distance between the $P_k$ distributions as a performance measure. The classic method for measuring the distance between two probability distributions is the Kulback-Leiber (KL) divergence, and it is defined as

$$D_{KL}(P_{k_1}(y), P_{k_2}(y)) = \int_y P_{k_1}(y)(\log P_{k_1}(y) - \log P_{k_2}(y)) \cdot dy, \tag{4.34}$$

where $P_{k_1}$ and $P_{k_2}$ are the probability distributions for $f_k$ at scales $k_1$ and $k_2$ respectively. By substituting in equations 4.32 and 4.33, the above definition of the KL divergence can be simplified to:

$$D_{KL}(P_{k_1}(y), P_{k_2}(y)) = \sum_{v,\omega}\left(\frac{\sigma_{k_1}(v,\omega)}{\sigma_{k_2}(v,\omega)} - \log\frac{\sigma_{k_1}(v,\omega)}{\sigma_{k_2}(v,\omega)}\right). \tag{4.35}$$

The above divergence is maximized for a PSF that has frequencies $(v,\omega)$ where the ratio between $F_{k_1}(v,\omega)$ and $F_{k_2}(v,\omega)$ is large. This agrees with our intuitive proposal that stated

that a high performing aperture shape will have unique zero positions for different scales. It is also interesting to note that equation 4.35 is dependent only on $\sigma(v, \omega)$, which is in turn dependent on $F_k$, $\eta$, and $\alpha$. This suggests that the performance of a particular aperture shape is independent of scene specifics (which would be represented in the ideal sharp image $x$), as long as the scene conforms to the sparse derivative prior assumed initially. However, as was seen in the case of defocus deblurring, the performance of an aperture shape is highly dependent on the level of noise present, and will therefore be affected by camera parameters and lighting conditions.

Finally, in order to construct a principled performance criterion, the KL divergence in equation 4.35 is calculated between various scales of a particular aperture shape and the minimum divergence between any two scales is recorded [18]. A low divergence value indicates that the two scales cannot be easily distinguished, and therefore the minimum divergence amongst the various scales represents the worst-case scenario for that particular aperture shape. The KL divergence must be taken for each possible scale pair, for example if three scales were tested, then the final fitness function would be

$$R(f_k) = min[D_{KL}(P_{k_1}, P_{k_2}), D_{KL}(P_{k_1}, P_{k_3}), D_{KL}(P_{k_2}, P_{k_3})]. \tag{4.36}$$

For the sake of accuracy, the scales tested should cover the entire range necessary for the expected depths in the scene, and the increments between each scale should be as fine as possible. Unfortunately the number of unique scale parings increases with the factorial of the total number of scales, and therefore the number of scales tested becomes a compromise between accuracy and computational complexity.

**Searching for the Optimum Aperture**

As in the case of searching for the optimal aperture shape for defocus deblurring there are $2^{N \times N}$ unique combinations for an $N \times N$ binary matrix, and it is therefore infeasible to perform an exhaustive search through the entire solution space looking for the global optimum. Instead, the same genetic algorithm described in section 4.2 (detailed in figure 4.8) was used again to intelligently search for an optimal shape. The only change made was to replace the deblurring fitness function with the one just derived for depth discrimination.

### 4.3.3 Blur Scale Identification

The probability model $P_k(y)$ (equation 4.32) gives the probability that a given observed image was blurred with a particular PSF blur scale, and therefore the maximum likely blur scale $\hat{k}$ could be estimated by determining the blur scale $k$ that maximizes $P_k(y)$:

$$\hat{k} = \arg\max_k [P_k(y)]. \tag{4.37}$$

Unfortunately this method has been found to be unreliable when faced with practical problems such as high-frequency noise [18]. Instead, a more robust method has been suggested that uses an unnormalized energy term $E_k(y)$:

$$E_k(y) = y^T \Sigma_k^{-1} y, \tag{4.38}$$

where $\Sigma_k$ was defined in equation 4.31 as the linearly transformed version of the prior covariance matrix $\Psi_k$. The estimated blur scale $\hat{k}$ will then be the scale $k$ that minimizes $E_k(y)$ (in conjunction with a scale dependent scalar weighting $\lambda_k$):

$$\hat{k} = \arg\min_k \lambda_k E_k(y), \tag{4.39}$$

where the weighting vector $\lambda_k$ must be independently tuned using a set of training images to minimize the rate of misidentification. Unfortunately evaluating $E_k(y)$ is slow since it requires the inversion of large matrices, and therefore in practice it can be approximated by [18]:

$$y^T \Sigma_k^{-1} y \approx \frac{1}{\eta^2} |C_{f_k} \hat{x}_k - y|^2, \tag{4.40}$$

where $\hat{x}_k$ is the observed image deblurred with the PSF at scale $k$, and can be calculated using equation 4.23 from section 4.2. The approximation for $E_k(y)$ above calculates the reconstruction error by re-blurring $\hat{x}$ using the PSF at the scale under investigation and then comparing it to the observed image $y$. The approximation relies on the assumption that refocusing followed by re-blurring using an incorrect blur scale will produce significant ringing artifacts and therefore the reconstruction error will be large.

Therefore, for an observed blurry image $y$ of unknown depth, $E_k(y)$ (or its approximation) is calculated for each PSF scale $k$ in the specified range, and the scale that minimizes $\lambda_k E_k$ is chosen as the maximum likelihood solution.

## 4.4 Light Fields with Coded Apertures

### 4.4.1 Capturing the Light Field

The concept of the 4D light field was introduced in section 2.1, and was defined as a function which returns the intensity of a light ray passing through a particular point on a 2D surface in a particular angular direction. Various methods for physically capturing a light field were also presented, such as using an array of cameras, or placing additional optics within the camera itself. In this section a method for capturing a light field using coded apertures is presented. Our method is based on the method described by Liang et al. [21].

The representation used for the captured light field is the two-plane representation shown in figure 2.4 (c), where a ray is defined by its intersection with the two planes $(u, v)$ and $(s, t)$. As shown in figure 4.13, the model used in this method places the $(u, v)$ plane at the aperture plane and the $(s, t)$ plane at the image plane. This means that $(s, t)$ coordinates map directly to $(x, y)$ pixel coordinates on the 2D image sensor, while $(u, v)$ coordinates map to coordinates $(i, j)$ on the coded aperture shape, $f_k(i, j)$.

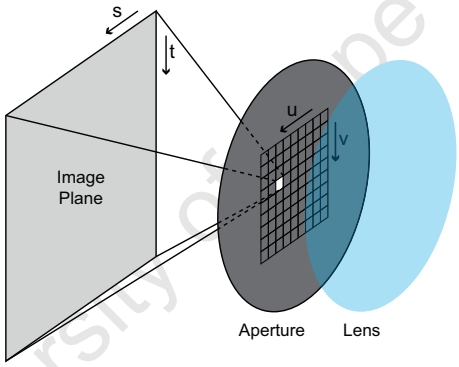



Figure 4.13: Diagram showing how a light field can be captured by taking a sequence of photographs, each with a different aperture block open at a time. By constraining the $(u, v)$ coordinate the 2D image obtained represents a single slice of the 4D light field. By combining multiple slices the full light field can be reconstructed.

A conventional aperture allows rays to pass through many locations on the $(u, v)$ plane, and each ray may reach the image plane at any $(s, t)$ location. This 4D function cannot be captured by the 2D image sensor, and the light rays are instead integrated over their angular dimensions and information is lost. However, if a coded aperture that has a single open block is used instead, only rays passing through a single position on the $(u, v)$ plane are allowed to reach the image plane, and therefore the 2D image formed represents a 2D slice (with a constant $(u, v)$ coordinate) of the larger 4D function. If multiple photographs are taken, each with a different $(i, j)$ aperture block open at a time, then the full light field can be reconstructed from the multiple slices [21]. This method therefore corresponds to the

model shown in figure 2.6 (a), in which each $(u, v)$ coordinate represents a unique captured image, and each $(s, t)$ coordinate represents a particular pixel in that image.

This coded aperture method differs most significantly from the other single-camera methods described in section 2.1 (micro-lens array and cosine mask) in that it does not sacrifice spacial resolution in order to capture angular resolution. Instead, by requiring multiple photographs of the same scene it trades temporal resolution for angular resolution. While this means that this method is not suitable for moving scenes, it also allows high resolution light fields to be captured. Also, the amount of angular resolution captured can be reconfigured at any time by simply adjusting the resolution of binary coded aperture. An $N \times N$ binary aperture has $N^2$ individual blocks, and a photograph must be taken with each block to capture the full light field, and so the number of required photographs increases as the angular resolution becomes finer.

### 4.4.2 Stereo Disparity

A simple but useful application for a captured light field is to calculate depth from stereo pixel disparity. As shown in figure 4.14, a slightly different view of a scene is obtained for each $(u, v)$ value in the light field. The relationship between each view is well defined since it depends only on the intrinsic parameters of the lens and the geometry of the coded aperture used to capture the light field. This fact greatly simplifies the image rectification step required for stereo disparity calculation, which conventionally requires accurately measured extrinsic camera parameters. There are $\binom{N^2}{2}$ available stereo pairs in a light field with a $(u, v)$ resolution of $N \times N$, and the stereo disparity result obtained from each pair can merged together to reject outliers and generate multi-view 3D point clouds [20].

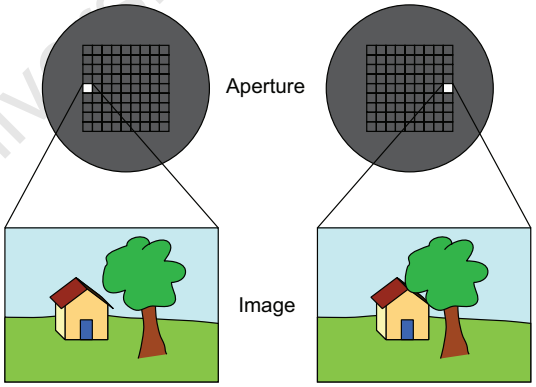



Figure 4.14: Diagram showing how coded apertures can produce image pairs suitable for stereo disparity methods. Light rays passing through each aperture block produce slightly different views of a scene without requiring that the camera be physically moved.

### 4.4.3 Synthesizing Virtual Photographs

Another popular application for a captured light field is to synthesize virtual photographs. The parameters of these virtual photographs can differ from those used in the original photographs, and this allows a user to adjust parameters such as the shape of the aperture, or the position of the focal plane without having to take any additional photographs.

As mentioned earlier, by placing the $(u, v)$ plane of the light field at the aperture plane, the shape of the aperture only allows light rays to pass through a specific subset of $(u, v)$ locations. Therefore, since the rays passing through any $(u, v)$ location can be determined from a captured light field, any aperture shape can be synthesized by integrating the light field over the appropriate $(u, v)$ subset [21]. Figure 4.15 graphically depicts this integration process.

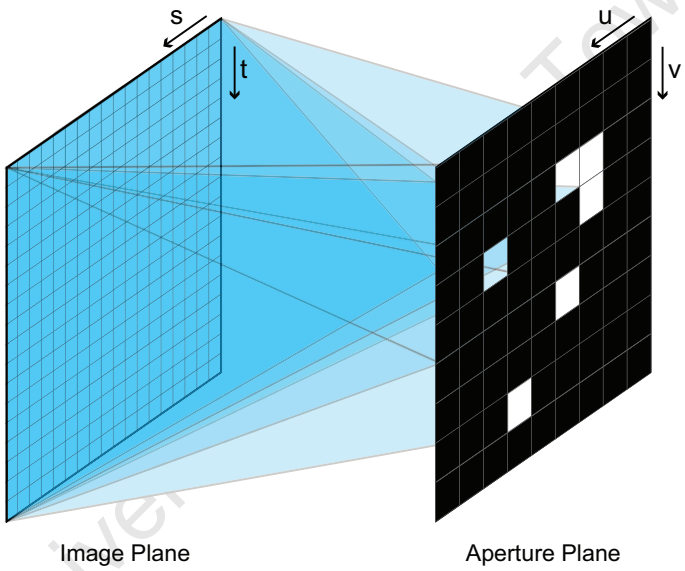



Figure 4.15: Diagram showing how a light field allows a virtual photograph to be synthesized with any aperture pattern. By integrating the 4D light field over a specific subset of $(u, v)$ values, a 2D image is produced that is equivalent to a real photograph taken with the corresponding aperture shape.

The position of the focal plane in a synthesized photograph can be adjusted by inserting a virtual image plane into the standard light field model. The position of the virtual image plane is determined by the imaging equation for thin lenses:

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f}, \tag{4.41}$$

where $f$ is the focal distance of the lens, and $S_1$ and $S_2$ are the distances from the focal

plane to the lens, and from the lens to the image plane respectively. Placing the virtual image plane further away from the lens than the original image plane will form a virtual photograph with a focal plane nearer to the lens, while placing the virtual image plane closer to the lens will form a virtual image with a focal plane further away from the lens. In this way any object in the scene can be brought into focus, regardless of its depth.

Since the inside of a camera can be assumed to be free-space, calculating the image formed at a virtual image plane can be performed by tracing the rays defined by the 4D light field and determining their points of intersection with the virtual plane. Figure 4.16 shows the standard two plane light field model, with the addition of a virtual image plane placed a distance of $\alpha$ away from the $(u, v)$ plane. For simplicity all the other dimensions (i.e. the distance from the original image plane to the aperture plane, as well as the overall dimensions of the planes themselves) have been normalized to a value of 1.0. The figure also shows a single light ray passing through the $(u, v)$, $(s, t)$ and $(s', t')$ planes, intersecting with the planes at points $(x_0, y_0)$, $(x_1, y_1)$ and $(x_2, y_2)$ respectively.

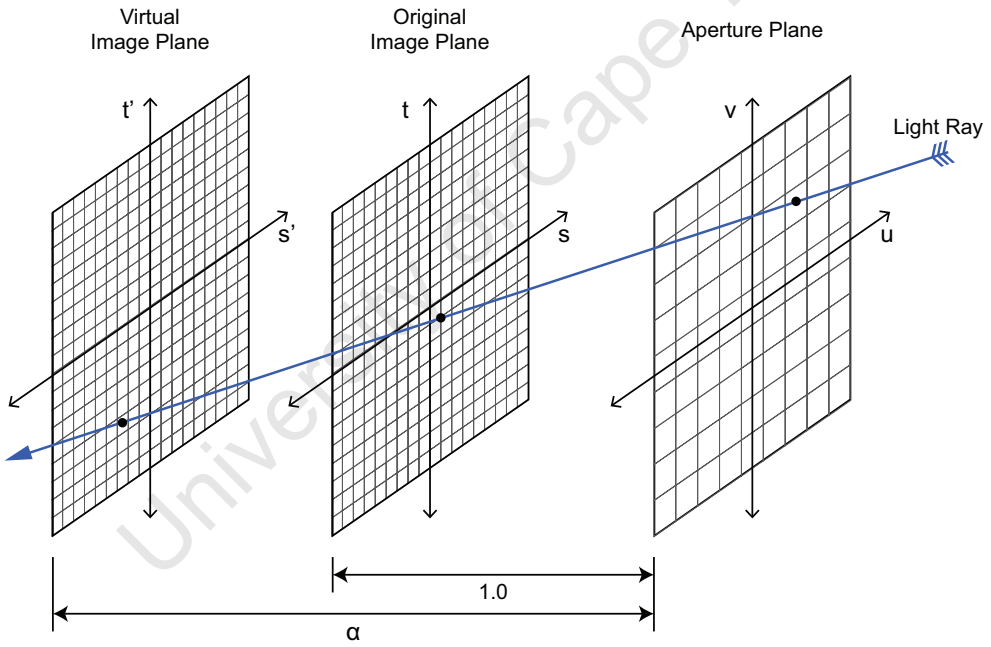



Figure 4.16: Diagram showing how a refocused virtual photograph can be synthesized from a 4D light field. The image formed at the virtual image plane $(s', t')$ is calculated by geometrically transforming the 4D light field located at the real image plane $(s, t)$. The focus is adjusted by moving the virtual image plane closer or further away from the aperture plane $(u, v)$.

Assuming that the three planes are parallel and are centered on the optical axis, the points of

intersections between the planes and the light ray can be related by the following geometric transformations:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \tag{4.42}$$

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - \alpha \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \tag{4.43}$$

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + (\alpha - 1) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}, \tag{4.44}$$

where $\Delta x$ and $\Delta y$ are the vertical and horizontal gradients of the light ray. Since the radiance of the ray does not change along its length, we also know that the value of the light field at each point of intersection is equal:

$$L_{s',t',u,v}(x_2, y_2, x_0, y_0) = L_{s,t,u,v}(x_1, y_1, x_0, y_0) \tag{4.45}$$

$$= L_{s,t,u,v}(x_2 + (\alpha - 1)\Delta x, y_2 + (\alpha - 1)\Delta y, x_0, y_0), \tag{4.46}$$

where $L_{s',t',u,v}$ and $L_{s,t,u,v}$ represent the light field defined by planes $(s', t'), (u, v)$ and $(s, t), (u, v)$ respectively. Since $L_{s,t,u,v}$ is known (it is the original representation of captured light field), equation 4.45 allows us to calculate the light field at the virtual plane, following which the virtual image can be calculated by integrating the rays over the $(u, v)$ dimensions.

# Chapter 5

# Coded Exposure Methods

This chapter covers the theory of coded exposure photography, and its application to motion deblurring. The first section introduces the concept of motion blur by describing the role of a shutter in conventional exposures, and then explains how under certain conditions, motion blur can be modeled by convolution with a point-spread function (PSF). The second section describes how motion blur can be reduced through deconvolution, and explains how coded exposures overcome the difficulties associated with deblurring traditional exposure photographs. Lastly, two performance metrics are developed, and a method for searching for optimal exposure patterns using a genetic algorithm is presented.



## 5.1 Conventional Exposure

An ideal camera would be able to record an image formed by incoming light rays at a single instant in time, but unfortunately this is not physically possible since the measurement of light intensity is fundamentally an integrative process over time. Therefore all real cameras require a finite-length exposure time, in which the light rays are allowed to illuminate the image sensor until a well-exposed image is produced with an acceptable signal-to-noise ratio (SNR). The exposure time is controlled by the camera's shutter, which is a physical barrier that is placed between the lens and the image sensor and only allows light rays to enter the camera for a specific length of time. Figure 5.1 shows the standard thin-lens camera model with a closed shutter.

A finite exposure time does not affect the image produced of a stationary scene, since the incoming rays do not change over time and therefore reach the image plane at constant locations. However, if there is significant relative motion between the camera and the scene, the incoming rays change over the exposure time and illuminate a range of locations on the image plane, which 'smears' the true image in the direction of movement. This causes the final image to look blurred in the direction of movement, and is known as *motion blur*. Conven-

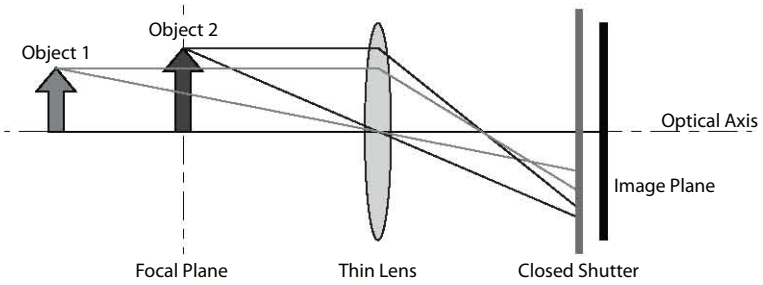



Figure 5.1: Diagram showing the thin-lens camera model with added physical shutter between the lens and the image plane. An ideal shutter blocks all rays when closed, allows all rays to pass when open, and can change state instantaneously.

tional photography assumes a $0^{th}$ order motion model (stationary scene), and the exposure time is shortened until the effect of any scene motion becomes unnoticeable. Unfortunately in some cases shortening the exposure time is impossible (e.g. in low light conditions) and so motion blur remains a problem [29]. Figure 5.2 compares the case of a stationary object being imaged (a), with that of a moving object (b) in a one-dimensional camera system. In (a) the scene object is mapped onto a constant set of pixels, thereby forming a well defined image, while in (b) the moving object illuminates a range of pixels. The size of the blur is proportional to both the velocity of the motion and the length of exposure.

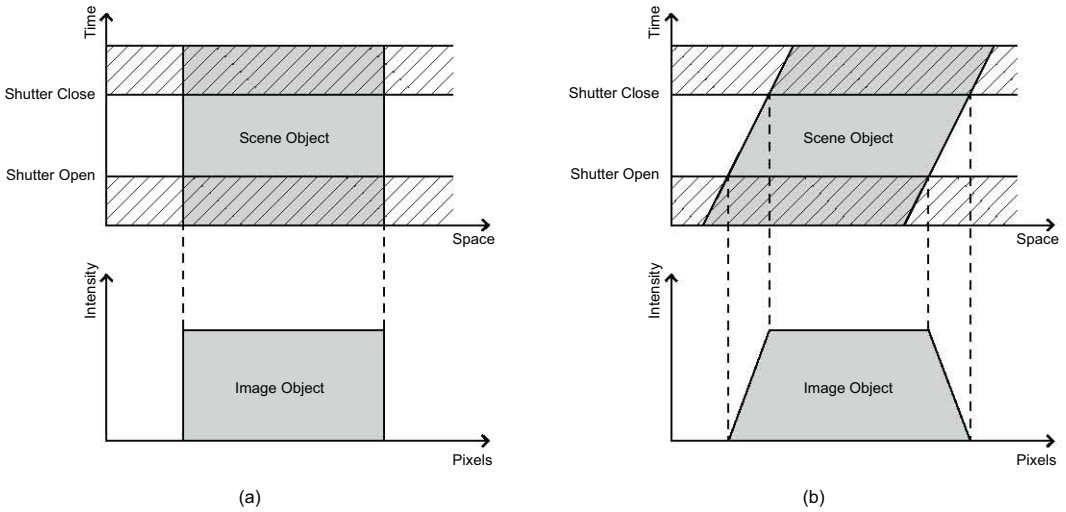



Figure 5.2: Diagram showing the problem of motion blur in a one-dimensional system. In (a) a stationary object (i.e. not moving in space over time) is imaged over the finite exposure time and produces a well defined image. In (b) a moving object is imaged over the same exposure time, and its edges are smeared across a number of pixels. Diagram inspired by [29].

As was the case with defocus blur, motion blur can also be modeled as a convolution between the true sharp image and a point-spread function (PSF), provided that the view of the scene object does not change significantly during its motion. If a moving point source was photographed with a stationary camera, the light rays originating from the source would trace a 2D projection of the point's motion on the image sensor. The shape of this trace would be determined by the direction component of the point's velocity, and the length of the trace would be determined by the magnitude of the point's velocity as well as the length of exposure [29]. This concept is graphically depicted in figure 5.3.

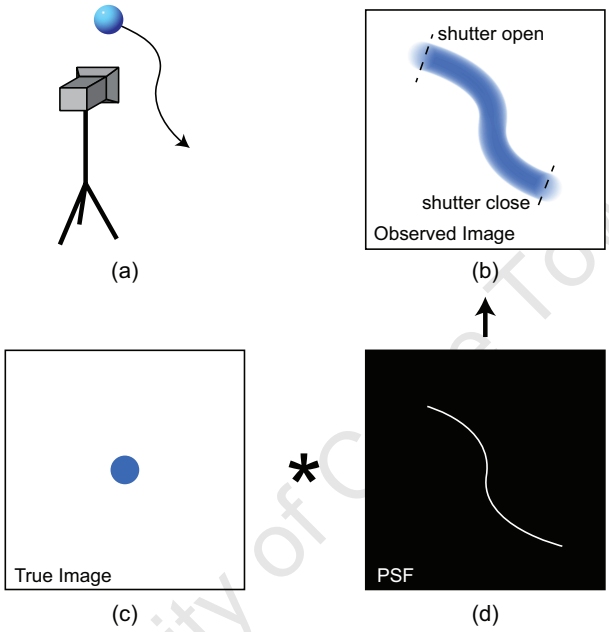



Figure 5.3: Diagram showing how motion blur can be modeled as convolution: (a) a scene with a moving object, (b) the resulting photograph containing motion blur, (c) the true sharp image, and (d) the motion-dependent PSF.

## 5.2 Reducing Motion Blur with Coded Exposure

### 5.2.1 Problem Overview

As stated in the previous section, provided that the view of a moving scene does not change significantly during its motion (i.e. no rotation or perspective effects), motion blur in a photograph can be modeled as the convolution between the true sharp image of the scene and a motion-dependent PSF. Therefore, if the PSF can be estimated (requiring knowledge of the scene's motion and the exposure time used to capture the image), removing motion blur becomes a standard deconvolution problem, just as in the case of defocus deblurring (section

4.2). In fact, besides estimating the PSF, the problem of motion deblurring is identical to that of defocus deblurring

As shown in figure 5.4 (a), a conventional exposure can be depicted as a one-dimensional rectangular pulse over time, and therefore the magnitude of its frequency spectrum takes the form of the sinc function. Since convolution in the time domain is multiplication in the frequency domain, when a sharp image is blurred with a conventional exposure, any frequency information located at the PSF's multiple zero positions is irreversibly lost. This makes motion deblurring an ill-posed problem [29]. However, just as the conventional aperture in section 4.2 could be replaced with a coded binary aperture, so too can the conventional pulse-shape exposure be replaced with a coded binary exposure. Figure 5.4 (b) shows how a random binary exposure can preserve frequency information far better than a conventional exposure.

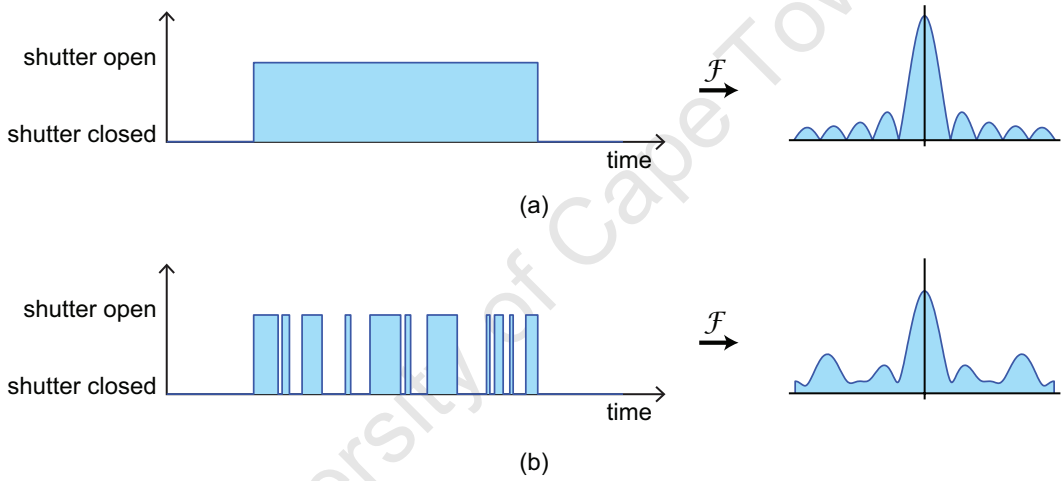



Figure 5.4: Figure comparing the one-dimensional frequency spectrums of: (a) a conventional pulse exposure, and (b) a random binary exposure.



### 5.2.2 Selecting a Coded Exposure Pattern

#### Exposure Pattern Representation

Since a binary coded exposure pattern is one-dimensional, it can simply be represented by a binary vector $f_k$ of length $N$:

$$f_k = \left[ \begin{array}{cccc} b_1 & b_2 & \dots & b_N \end{array} \right]. \tag{5.1}$$

**Exposure Pattern Performance Metric**

Before an optimal exposure pattern can be found, a metric needs to be developed that can be used to rate the performance of a particular exposure pattern. Intuitively, a well performing pattern should preserve as much frequency information as possible, and so will have large minimum values in its frequency spectrum. This is the metric that was used by Raskar et al. [29]. In addition, they propose that a well performing pattern should have a low magnitude variance in its frequency spectrum, since this will reduce the artifacts produced if the motion blurring PSF is estimated slightly incorrectly. Therefore, the metric that they use takes the following form:

$$R_{raskar}(f_k) = \alpha \cdot min(|F_k|) + \beta / variance(|F_k|) \tag{5.2}$$

where $F_k$ is the Fourier transform of $f_k$, and $\alpha$ and $\beta$ are tunable scalars that determine the weighting of the two components of the metric. The optimal exposure pattern $\hat{f}_k$ is the value of $f_k$ that maximizes $R_{raskar}(f_k)$:

$$\hat{f}_k = \arg\max_{f_k}(R_{raskar}(f_k)) \tag{5.3}$$

While the performance metric defined by $R_{raskar}(f_k)$ is simple to calculate, it does not take into consideration any properties of natural images or the level of noise present, and therefore it may not be the optimal metric to use with motion deblurring of real photographs. In order to compare the suitability of $R_{raskar}(f_k)$, a second metric $R_{zhou}(F_k)$, based on the $R(F_k)$ function defined in equation 4.21 (section 4.2), was also used to find an optimal exposure pattern. The only major alteration was to redefine the original two-dimensional function as a one-dimensional function:

$$R_{zhou}(F_k) = \sum_{\omega} \frac{\sigma^2}{|F_k(\omega)|^2 + \frac{\sigma^2}{A(\omega)}}, \tag{5.4}$$

where $\omega$ is the frequency variable, $\sigma$ is the standard deviation of the assumed Gaussian noise, and $A(\omega)$ is an approximation of the power spectra of natural images. For a cursory discussion of the derivation of this metric see section 4.2, and for a comprehensive coverage see Zhou and Nayar's original paper [39]. $R_{zhou}(F_k)$ provides a measure of the amount that noise will be amplified by a particular pattern, and so the optimal deblurring exposure pattern $\hat{f}_k$ will be the value of $f_k$ that minimizes $R_{zhou}(F_k)$:

$$\hat{f}_k = \arg\min_{f_k}(R_{zhou}(f_k)). \tag{5.5}$$

#### Searching for the Optimum Exposure Pattern

For a binary vector of length $N$ there are $2^N$ possible combinations, and so an exhaustive search of the solution space is infeasible. Therefore a genetic algorithm was used to intelligently search through the solution space for an optimal pattern. The genetic algorithm used was a one-dimensional version of the algorithm described by figure 4.8 in section 4.2. The only other change was to replace the fitness function with the performance metrics defined earlier in equations 5.2 and 5.4.

### 5.2.3 Motion Deblurring

#### Estimating the PSF

The first step in motion deblurring is to estimate the PSF. Under the assumption of constant linear motion, the PSF will always be a straight line with a width of 1 pixel and a length of $n$ pixels. The angle of the PSF will be equal to the angle of object motion (projected onto the image sensor), and the intensity value of the PSF along the line will be modulated by the coded exposure pattern used. The length of the linear PSF can be calculated from the scene geometry and intrinsic camera parameters, but for simplicity it can also be calculated directly from the extent to which a sharp object is elongated by the blur. This second method requires knowing the size of an identifiable object in the sharp image, and therefore it cannot be used in all situations. Since the blurred image was formed from a convolution between the sharp image and the PSF, the pixel length, $p$, of a blurred image object in the direction of blurring will be given by $p = m + n - 1$, where $m$ is the length of the sharp image object and $n$ is the length of the linear PSF (both in pixels). Rearranging to make the length of the PSF the subject of the equation we get

$$n = p - m + 1. \tag{5.6}$$

Figures 5.5 and 5.6 show a graphic example of how the shape and size of a motion blur PSF can be estimated using the methods described above.

#### Deconvolution

The second step in the motion deblurring is deconvolution using the PSF estimated above. Even with a PSF with no zeros in its frequency spectrum, the task of deconvolution is still sensitive to noise and errors in the estimated PSF. This has lead to the development of many different algorithms, any of which could be used for motion deblurring. Since an optimized PSF should improve the results obtained regardless of which algorithm is used, the choice of algorithm is largely inconsequential. However, for consistency it was decided to use the same algorithm that was used for defocus deblurring in section 4.2.

The deconvolution function is defined in equation 4.23, and it uses a statistical prior that assumes that the derivatives of the sharp image follow a sparse distribution. This has been shown to be a good assumption for natural images, which have smooth gradients over most of the image and a small number of large gradients located at positions such as object boundaries. For a brief discussion of this deconvolution algorithm see section 4.2, and for a thorough analysis see its derivation in the paper by Levin et al. [18].

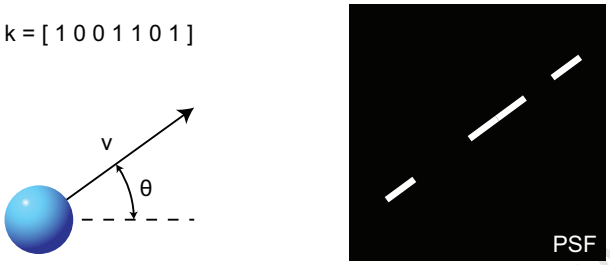



Figure 5.5: Diagram showing how the shape of the motion blur PSF can be estimated from the binary coded exposure pattern and the direction of object motion. For linear motion the PSF will always be a straight line modulated by the binary pattern, and its angle will be equal to the angle of motion.

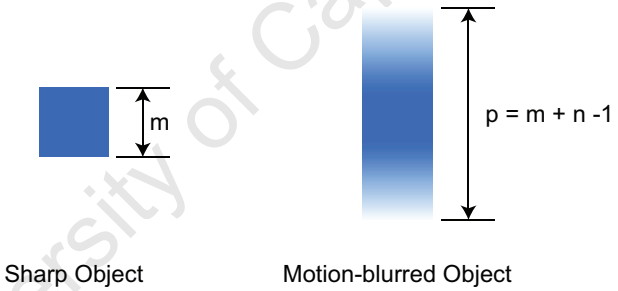



Figure 5.6: Diagram showing how the size of the motion blur PSF can be estimated from the extent to which a sharp object is elongated. If the sharp object is $m$ pixels long in the direction of motion and the blurred object is $p$ pixels long, then the length of the PSF $n = p - m + 1$.

# Chapter 6

# Experimental Procedure

## 6.1 Camera Parameters

Unless stated otherwise, the following camera parameters were used for all the experiments performed in this report:

| | |
|---|---|
| **Image Dimensions** | 792 × 1188 pixels |
| **Image Color Mode** | 16-bit, monochrome |
| **Image File Format** | Portable Network Graphic (PNG) |
| **ISO Speed** | ISO 100 |
| **Lens Focal Length** | 50mm |

### 6.1.1 Noise Measurement

The first camera parameter that needed to be measured was the level of noise in the captured images. In every coded photography method that has been described so far, noise has been an important consideration in the models. Therefore an accurate estimate of the noise power is required before any other experiments can be performed.

There are many different types of noise in digital cameras, and each type can originate from a number of different sources. Some types, such as read noise, dark noise and quantization noise are caused by physical and electrical processes and are independent of the light signal being captured, while others such as shot noise are produced by the quantum nature of photons and therefore vary with the intensity of incoming light signal [24]. A rigorous noise analysis would require identifying each source of noise and modeling it in isolation. However, performing this kind of analysis accurately can become complicated, and in our particular situation such detailed information would be of no extra use. Instead, we model the combined effect of all the different kinds of noise as a single additive white Gaussian

process, since this has been the assumption made in all the coded photography methods that have been described in this report.

Under the assumption of zero-mean, independent and identically distributed (IID) additive white Gaussian noise, the only parameter that must be measured is the standard deviation of the distribution. It has also been assumed that the total sensor integration time, the average temperature, and the average scene illumination intensity all remain relatively constant, and therefore the only variable that will affect the level of noise is the sensor gain setting. The sensor gain for our prototype camera is determined by the ISO speed that has been selected, and so a noise measurement was made for each of the following settings: ISO 100, ISO 200, ISO 400 and ISO 800.

Measuring the standard deviation of the noise was performed by taking multiple photographs (10 in our implementation) of a gray planar target at each ISO setting. Care was taken to illuminate the target uniformly and the lens was set to the maximum level of defocus in order to remove any high frequency texture that may have been present in the target. By making the target completely uniform, it could be assumed that any intensity deviation that existed between pixels in the captured images was caused solely by the additive noise, and so the standard deviation of the Gaussian noise could be approximated by the sample's standard deviation. Furthermore, to remove any low-frequency optical effects (such as lens vignetting) that may have been present, the images were divided into small patches ($32 \times 16$ pixels in our implementation). The standard deviation was calculated for each patch in isolation, and then an average value was calculated across all the patches and across all the images for each ISO setting. The average standard deviation values that were obtained are listed in the table below:

| **Sensor Gain** | **ISO 100** | **ISO 200** | **ISO 400** | **ISO 800** |
|---|---|---|---|---|
| **Std. Dev.** | 0.0084 | 0.0112 | 0.0154 | 0.0223 |

The images were all monochrome and were represented by 16-bit intensity values ranging between 0 and 1. Therefore, if we assume a signal intensity of 0.5 (50%), the standard deviations listed above can be represented as the following signal-to-noise (SNR) power ratios:

| **Sensor Gain** | **ISO 100** | **ISO 200** | **ISO 400** | **ISO 800** |
|---|---|---|---|---|
| **Std. Dev.** | 35.5 dB | 33.0 dB | 30.2 dB | 27.0 dB |

Unsurprisingly, the lowest noise level was obtained when the sensor speed was set at ISO 100, which represents the lowest level of sensor gain. Since every one of the coded photography methods that have been discussed in this report perform better with a lower level of noise, all the experiments that are described later in this chapter were performed with the camera speed set to ISO 100.

## 6.2 Coded Aperture Experiments

Figure 6.1 shows a block-diagram that provides an overview of the coded aperture experimental procedure. The first step was to select the aperture shapes that would be used in the experiments, and to design and construct the test scenes. The selected aperture patterns were then calibrated, and together with the test scenes they were used to actually capture the experimental photographs. These captured images and the calibrated PSFs were then used to perform defocus deblurring and depth estimation. The rest of this section will be used to discuss each step of this process in more detail.

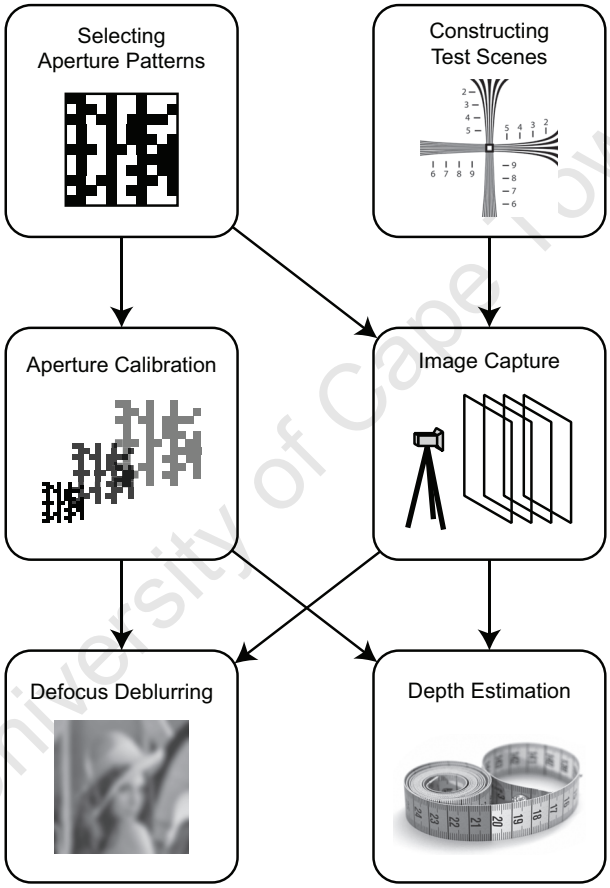





Figure 6.1: Block-diagram showing an overview of the coded aperture experimental procedure.

### 6.2.1 General Experimentation Configuration

For consistency, all the coded aperture experiments that were performed share a general configuration. In each experiment the prototype camera was loaded with a coded aperture made by cutting an $11 \times 11$ binary grid from a cardboard mask. The total size of this grid was set at 12mm $\times$ 12mm, and therefore the size of each individual block in the grid was 1.09mm $\times$ 1.09mm. The lens was then focused on a plane 1.0m away from the camera, and scene objects were placed at the center of the image frame, between 1.0m and 2.0m away from the camera. This setup is shown graphically in figure 6.2.

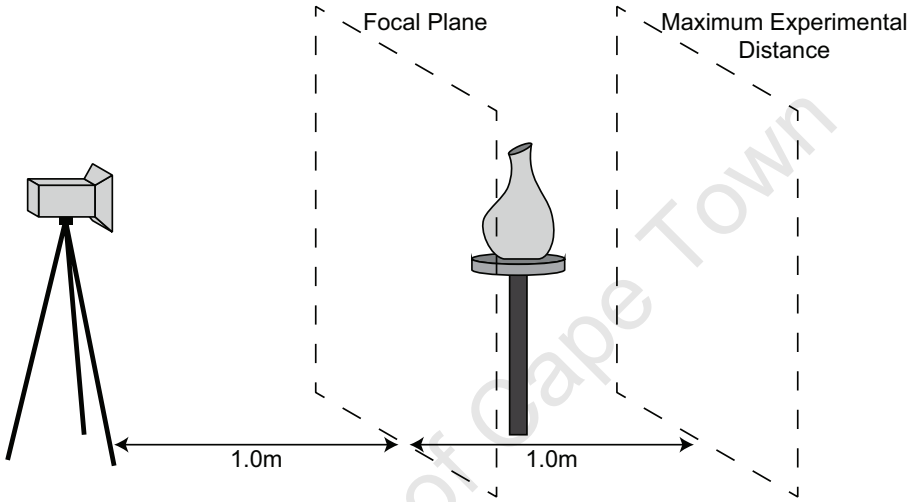



Figure 6.2: Diagram showing the general configuration used for all the coded aperture experiments described in this section. The camera lens was focused at 1.0m, and all scene objects were located between 1.0m and 2.0m from the camera.

For the configuration described, the minimum PSF scale occurs for objects located on the focal plane (1.0m away from the camera), and the maximum PSF scale occurs for objects located at the maximum experimental distance of 2.0m. To measure the size of the minimum and maximum PSF scales, a point light source was photographed at 1.0m and 2.0m. The resultant (cropped and enhanced) images are shown in figure 6.3. The figure shows that the minimum PSF size is about 3 pixels, while the maximum PSF is about 23 pixels in our input images (with pixel dimensions of $792 \times 1188$).

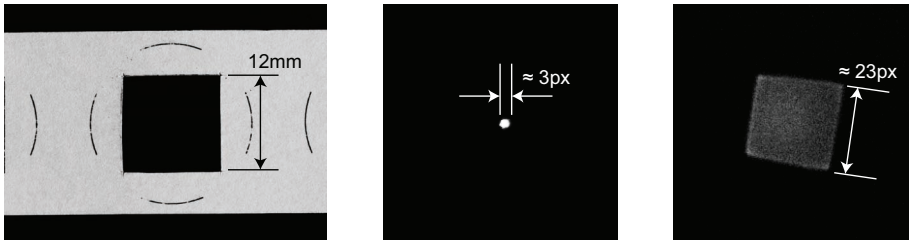



Figure 6.3: Photographs showing the minimum and maximum PSF sizes for the described experimental configuration. The minimum occurs at the focal plane 1.0m away from the camera, and the maximum occurs at the maximum experimental distance of 2.0m.

### 6.2.2 Selecting Aperture Patterns

The first stage in setting up the coded aperture experiments was to select the aperture shapes that would be used. In chapter 4 the theoretical methods for finding optimal deblurring and depth-discriminating apertures patterns were presented. In this section these methods are practically implemented, and the resulting patterns are displayed and discussed. For all the coded apertures selected, a square binary grid of size $11 \times 11$ ($N = 11$) was used since this was the largest value of $N$ that could be accurately manufactured.

#### Optimal Pattern for Defocus Deblurring

In section 4.2 a criterion, $R(F_k)$, was derived to rate the deblurring performance of a particular aperture (equation 4.21), and a genetic algorithm based method was described (figure 4.8) that could be used to optimize this criterion. Specifying $R(F_k)$ requires three parameters: the aperture under consideration $F_k$, the standard deviation of the noise in the observed image $\sigma$, and an approximation of the power spectrum of natural images $A(v, \omega)$. In our implementation $F_k$ is supplied by the genetic algorithm as it searches for the optimal pattern, and the value of $\sigma$ is taken from the noise measurement done in section 6.1 and is therefore set as 0.0084 (the standard deviation stated for a camera speed of ISO 100). However, the matrix representation of $A(v, \omega)$ still needs to be calculated.

We assume that the power spectra of natural images follow the $1/f$ rule, and this is expressed generally as

$$E[|X(f,\theta)|^2] = \frac{\beta(\theta)}{f^{\alpha(\theta)}}, \tag{6.1}$$

where $(f, \theta)$ are radial coordinates in the frequency domain, and $\alpha$ and $\beta$ are functions dependent on radial orientation $\theta$. Values for $\alpha$ and $\beta$ at horizontal, oblique, and vertical

orientations ($\theta = 0$, $\frac{\pi}{4}$, and $\frac{\pi}{2}$) were taken from Torralba and Oliva [34], and are listed in the table below:

| | **Horizontal** | **Oblique** | **Vertical** |
|---|---|---|---|
| $\alpha$ | 1.98 | 2.02 | 2.22 |
| $\beta$ | 0.96 | 0.86 | 1.0 |

Values for $\alpha$ and $\beta$ at all other orientations $\theta$ are calculated by linearly interpolating the values in the table, and the matrix $A(v, \omega)$ is calculated using equation 6.1. The resulting 2D matrix is shown in figure 6.4 as a surface in 3D space (with a log-scale $z$ axis). It clearly shows the characteristic 1/f curve which supposes that natural images have the majority of their content located at low frequencies.

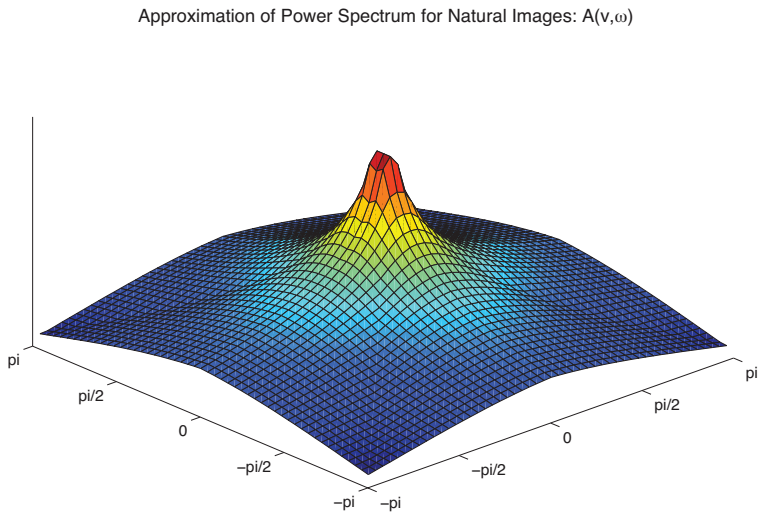



Figure 6.4: Surface plot of the matrix $A(v, \omega)$ that is used to approximate the power spectra of natural images. It is calculated using the common 1/f model.

The genetic algorithm also requires tuning parameters, namely: $G$ the maximum number of generations allowed, $S$ the total number of vectors in the initial population, $L$ the length of each vector (which is alway equal to $N \times N$), $M$ the number of best-performing vectors that survive each generation, $c1$ the probability of crossover, and $c2$ the probability of mutation. All of these values (other than $L$ which was set as $11 \times 11$) were adjusted manually until the genetic algorithm converged to an adequate solution in a reasonable amount of time. The final values are shown in the table below:

| **G** | **S** | **M** | **c1** | **c2** |
|---|---|---|---|---|
| 80 | 4000 | 400 | 0.2 | 0.05 |

Using the above parameters the genetic algorithm converges to a local minimum within 60 to 80 generations, as can be seen in figure 6.5, which shows the minimum value of $R(F_k)$ for each generation on a representative run. The algorithm was run five times with identical parameters, and the optimal aperture shape in each case is shown in figure 6.6. The value of $R(F_k)$ for each of the apertures is shown in the table below:

| | **(a)** | **(b)** | **(c)** | **(d)** | **(e)** |
|---|---|---|---|---|---|
| $R(F_k)$ | $1.505 \times 10^{-5}$ | $1.501 \times 10^{-5}$ | $1.504 \times 10^{-5}$ | $1.502 \times 10^{-5}$ | $1.509 \times 10^{-5}$ |

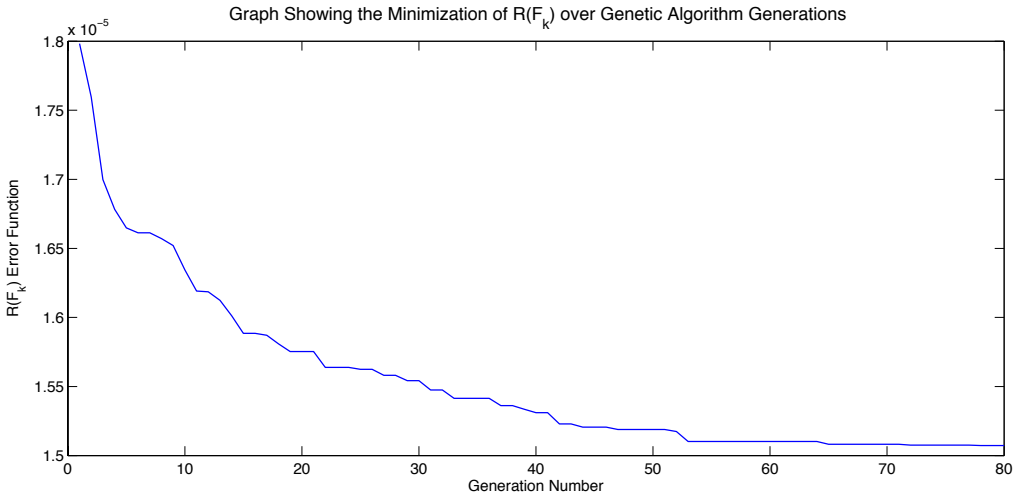



Figure 6.5: Graph showing the minimum $R(F_k)$ value for each generation of the genetic algorithm. The graph clearly shows that the algorithm converges to a local minimum after about 70 generations.

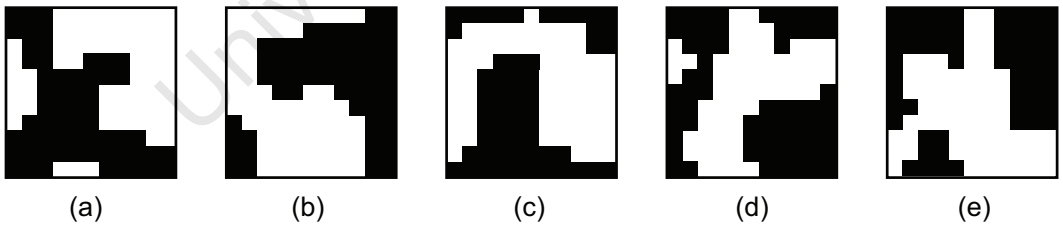



Figure 6.6: Image showing the output of five successive aperture optimizations using a genetic algorithm and the refocusing performance metric $R(F_k)$. While a different pattern was found each time, they all show similar structure. The pattern shown in (b) had the minimum $R(F_k)$ value, and so was selected for use in the refocusing experiments.

The fact that the five resultant aperture shapes are not identical indicates that the genetic algorithm converges to a local minimum rather than the global minimum. However, since the apertures display similar structure and since their $R(F_k)$ values are almost identical, it is likely that many near-optimal solutions exist within the solution space, and that the genetic algorithm is successfully converging on these solutions while avoiding the worse performers. Since pattern (b) in figure 6.6 has the lowest $R(F_k)$ value out of the five displayed, and since it can be physically manufactured by cutting the shape out of a mask material (i.e. there are no occluding areas completely surrounded by transparent areas), it was selected as the aperture that would be used in the defocus deblurring experiments that follow.

**Optimal Pattern for Depth Discrimination**

Aperture patterns were optimized for depth discrimination in the same way that they were optimized for refocussing, except that the depth discrimination fitness function was used instead of the deblurring fitness function. $R(f_k)$ was derived in section 4.3 (equation 4.36), and requires three parameters: the aperture $F_k$ at each of the scales under consideration, the standard deviation of the noise in the observed image $\eta$, and a scalar $\alpha$ that must be set to match the variance of the prior distribution to that of nature images. The value of $\alpha$ was set to 250 in our implementation, and the value of $\eta$ was again defined by the noise measurement specified in section 6.1. Eleven scales of the aperture $F_k$ were tested, ranging from 3 to 23 pixels in width, since this covered the scales expected from the particular experimental setup (as shown in figure 6.3).

The parameters of the genetic algorithm also required re-tuning for the new fitness function, but unsurprisingly the final values were similar to the values used for finding optimal deblurring apertures. The values used are listed in the table below:

| **G** | **S** | **M** | **c1** | **c2** |
|---|---|---|---|---|
| 80 | 1000 | 100 | 0.2 | 0.05 |

Figure 6.7 shows the maximum value of $R(f_k)$ for each generation on a representative run, and shows that the genetic algorithm successfully converges to a local maximum within 60 to 80 generations. Following the same procedure used for refocus optimization, the genetic algorithm was run five separate times, and the optimal aperture shape in each case is shown in 6.8. The value of $R(f_k)$ for each of the apertures is also shown in the table below:

| | **(a)** | **(b)** | **(c)** | **(d)** | **(e)** |
|---|---|---|---|---|---|
| $R(f_k)$ | $1.620 \times 10^4$ | $1.647 \times 10^4$ | $1.796 \times 10^4$ | $1.674 \times 10^4$ | $1.827 \times 10^4$ |

Again, the five non-identical aperture patterns produced by the genetic algorithm suggest that optimization converges to local maxima rather than the global maximum. However the five apertures all have similar $F(f_k)$ values, and their structure is visually similar. It is also

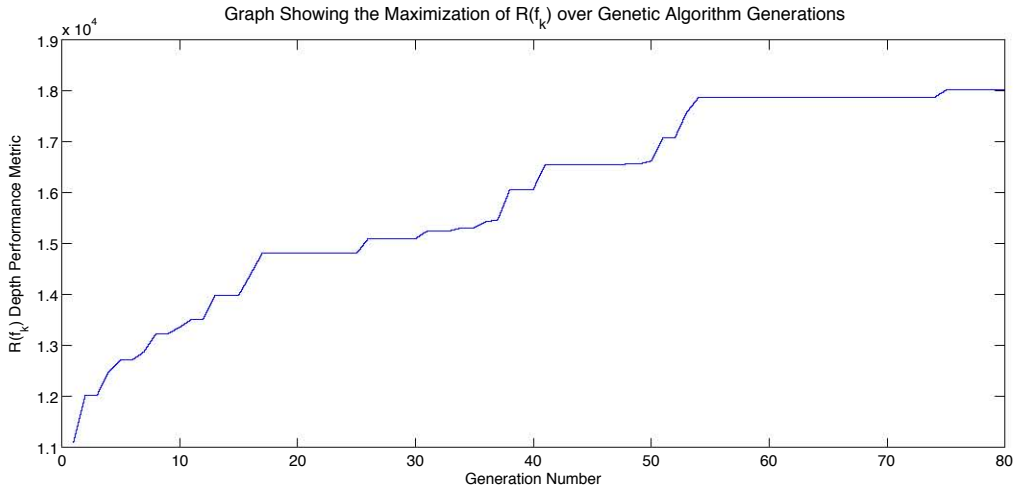



Figure 6.7: Graph showing the maximum $R(f_k)$ value for depth discrimination in each generation of the genetic algorithm. The graph shows that the algorithm converges to a local maximum after about 70 generations.

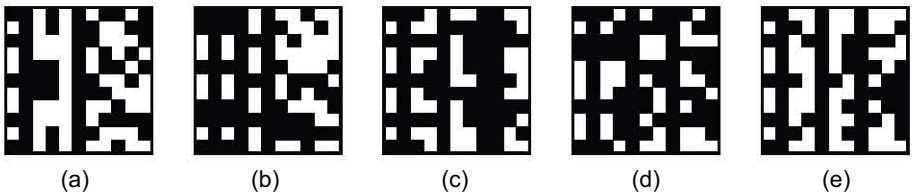



Figure 6.8: Image showing the output of five successive aperture optimizations using a genetic algorithm and the depth discrimination performance metric $R(f_k)$. While a different pattern was found each time, they all show similar structure. The pattern shown in (e) had the maximum $R(f_k)$ value, and so was selected for use in the depth estimation experiments.

interesting to note the significant level of symmetry that exists in all five of the apertures. Pattern (e) in figure 6.8 has the highest $R(f_k)$ value out of the five displayed and it can be physically manufactured, therefore it was selected as the aperture that would be used in the depth estimation experiments.

### Other Aperture Shapes for Comparison

In addition to the two optimized apertures that have been selected for refocussing and depth discrimination, the Coded Aperture experiments were also performed with a conventional round aperture, as well as the apertures developed by Levin [18] and Zhou [39] in their respective papers. The three additional apertures were included in order to compare their performance with that of the optimized apertures, and thereby to judge effectiveness of the optimization methods. These additional apertures can be seen in figure 6.9.

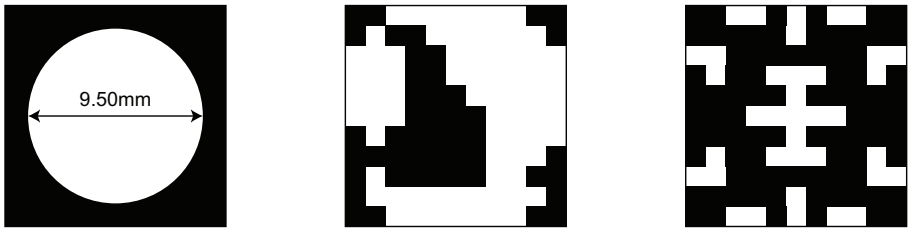



Figure 6.9: Diagram showing three additional aperture shapes that were used in experimentation: (a) a conventional round aperture, (b) the aperture shape selected by Zhou et al. [39], and (c) the aperture shape selected by Levin et al [18].

For fairness the conventional round aperture was designed to pass roughly the same amount of light as the two optimized apertures, so the diameter was chosen such that the total transparent area was equal to the average transparent area in the optimized apertures. The optimized apertures for refocusing (figure 6.6 (b)) and depth estimation (figure 6.8 (e)) have 62 and 57 transparent blocks respectively, and with each block having a physical area of $1.19\text{mm}^2$, this translates to an average transparent area of $70.8\text{mm}^2$. In order to have an equal area the round aperture was designed with a diameter of 9.50mm.

The aperture shown in figure 6.9 (b) was taken from Zhou and Nayar [39], and is included in order to compare with our optimized aperture for refocusing. The refocusing performance metric for apertures that is described in section 4.2 was also taken from their paper, and therefore their aperture shape looks similar to our optimal refocusing aperture shown in figure 6.6 (b). Zhou's aperture was optimized for a noise standard deviation of 0.008, which is close to the 0.0084 standard deviation measured in section 6.1.

The aperture shown in figure 6.9 (c) was taken from Levin et al. [18], and is included in order to compare with our optimized aperture for depth estimation. Levin's aperture was calculated using the same depth discrimination performance metric described in section 4.3, but with an extra constraint of enforcing horizontal and vertical symmetry. This constraint was imposed in order to dramatically reduce the number of possible shapes that had to be searched through, and was justified by the fact that symmetric aperture shapes seemed to perform better than non-symmetric apertures. It is due to this extra constraint that Levin's aperture differs so markedly from our depth optimized aperture shown in figure 6.8 (e).

### 6.2.3 Calibrating Apertures

In all the coded aperture methods discussed in chapter 4 it is assumed that the PSFs are known for each aperture at every distance from the camera. Although a PSF can be analytically estimated from an aperture's shape and the distance from the camera, this method does not consider construction defects or optical distortion. Instead, in order to account for such physical factors, the PSFs were estimated using empirical calibration. This required photographing a known calibration image at various distances using each aperture and then analyzing the captured images. The calibration image consisted of numerous circular points on a black background and it was displayed on an LCD monitor in a darkened room in order to approximate a collection of point light sources. For each aperture the lens was focused at 1.0m, and the calibration image was photographed every 10cm between 1.0m and 2.0m.

(a) (b)

Figure 6.10: An example image pair that was as input to the aperture calibration algorithm. (a) A sharp image of the calibration dots. (b) The observed blurred image.

The captured images were then cropped and centered, and the calibration image was registered to match the layout and scale of each observed image. The transformed calibration pattern and the observed image were then saved as a calibration pair, an example of which can be seen in figure 6.10. The transformed sharp calibration image, $X$, represents the input to a linear system, while the observed image, $B$, represents the associated output:

$$B = K * X, \quad \text{or} \quad \mathbf{b} = A\mathbf{k}, \tag{6.2}$$

where $K$ is the PSF, $\mathbf{b}$ and $\mathbf{k}$ are the vectorized versions of $B$ and $K$ respectively, and $A$ is the convolution matrix for $B$. Therefore the blur kernel, $\mathbf{k}$, can be estimated using a simple least-squares approximation

$$\mathbf{k} = [(A^T A + \lambda^2 I)^{-1} \cdot A^T] \cdot \mathbf{b}, \tag{6.3}$$

where the square-bracketed term represents the Moore-Penrose Inverse, and $\lambda$ is a tunable constant that adds Tikhanov regulation for stability. The maximum-likelihood estimate of $\mathbf{k}$ was calculated for each aperture shape at each distance, and stored for later use.

### 6.2.4 Test Scenes

Three different test scenes were used in the coded aperture experiments, and each was chosen to test the performance of the apertures under different conditions. For each scene the lens was focussed at 1.0m, and the scene was photographed every 10cm from 1.0m to 2.0m, repeating the process for each aperture tested. All the test scenes were photographed against an evenly illuminated white background.

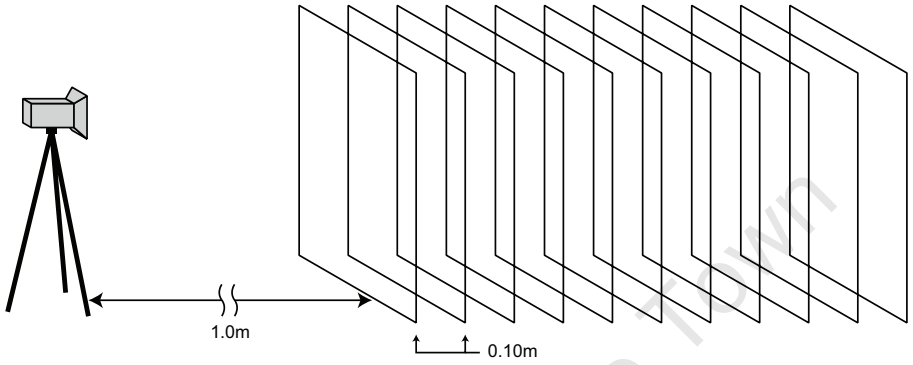



Figure 6.11: Figure showing how multiple photographs of each scene were taken from distances of 1.0m to 2.0m in 10cm increments. Diagram not drawn to scale.

The first test scene was a planar black and white diagram printed onto non-glossy paper and mounted onto a hard backboard. The diagram (shown in figure 6.12) was taken from the ISO 12233 resolution chart, which is the industry standard for measuring the resolving ability of optical systems. The diagram consists of a number of curved lines which are tightly spaced at the center of the image, and then slowly fan outward toward the edges of the diagram. The more tightly the lines are spaced, the more difficult it is to distinguish one line from another, and so the resolving ability of an optical system can be estimated by locating the point at which the converging lines blur into one another.

While the resolution diagram described is useful for accurately estimating the resolving performance of an optical system, it is not a good example of a natural image due to its high contrast and geometric structure, and therefore it is not the optimal test scene for apertures that were designed specifically for natural images. In order to address this issue the second test scene used in the coded aperture experiments was a human face. Humans are adept at noticing defects in images of the human face, and so it is an effective test scene for qualitative performance analysis. Since a human face is not planar, all depth measurements were made to the center of the eyes.

For both the aperture chart and the human face, only a small proportion of each captured

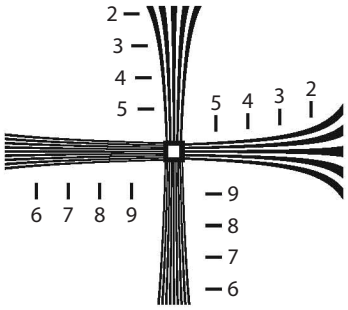



Figure 6.12: A diagram taken from the ISO 12233 chart and used as a test scene object in order to accurately estimate the performance of the coded apertures.

image was used to depict the scene, while the vast majority of each image was filled by the white background. Image areas filled by the flat background contain no high-frequency information, and so coded aperture methods that require that a significant proportion of the image power be located at high frequencies will not perform well for the two scenes described so far. Therefore a third scene was created that would completely fill the captured images with frequency-rich information. This scene consisted of a planar assortment of printed textual and photographic images, layered on top of one another in a complex arrangement. A view of this complex scene is shown in figure 6.13. Although not a natural scene, it does conform to the sparse derivative distribution prior assumed by the coded aperture methods.


Figure 6.13: Photograph showing a view of the complex planar scene.

### 6.2.5 Defocus Deblurring

Figure 6.14 shows an overview of the deblurring procedure. Firstly, the known aperture shape and scene distance are used to select the appropriate PSF from the set of calibrated PSFs. The PSF and the observed blurred image are then input into the deconvolution algorithm, which produces an estimated sharp image as output.

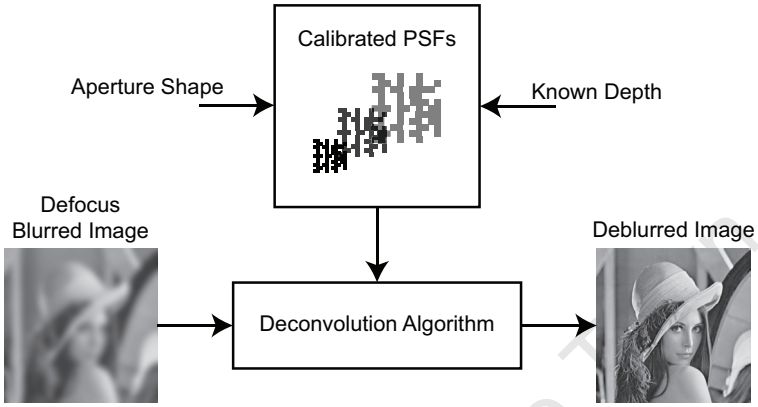



Figure 6.14: Block-diagram showing an overview the deblurring process.

The deconvolution algorithm used in the deblurring experiments was described in section 4.2 by equation 4.23. The algorithm was implemented by Levin et al. in MATLAB, and this was the implementation used in our experiments. Their function, *deconvSps(I,filt1,we,max_it)*, takes four input parameters, namely: the observed image $I$, the blur kernel $filt1$, a smoothing scalar $we$, and the maximum number of iterations allowed $max\_it$. The values of $we$ and $max\_it$ were set to 0.00025 and 100 respectively, and the algorithm took about two minutes to produce an output for our $792 \times 1188$ pixel input images.

### 6.2.6 Depth Estimation

Figure 6.15 shows and overview of the depth estimation procedure. Firstly, PSFs were taken from the calibrated PSF set that corresponded to the known aperture shape at each possible depth in the given range. Since the scene was photographed at depths ranging from 1.0m to 2.0m in 10cm increments, this was the range used for the calibrated PSFs. The selected PSFs, along with an observed blurred image, were then fed into two different depth estimation algorithms. One algorithm was based on the frequency domain zero-positions of the blurred image, while the other was based on reconstruction error. Both of these depth estimation methods were discussed in section 4.3, and are defined in equations 4.32 and 4.38.

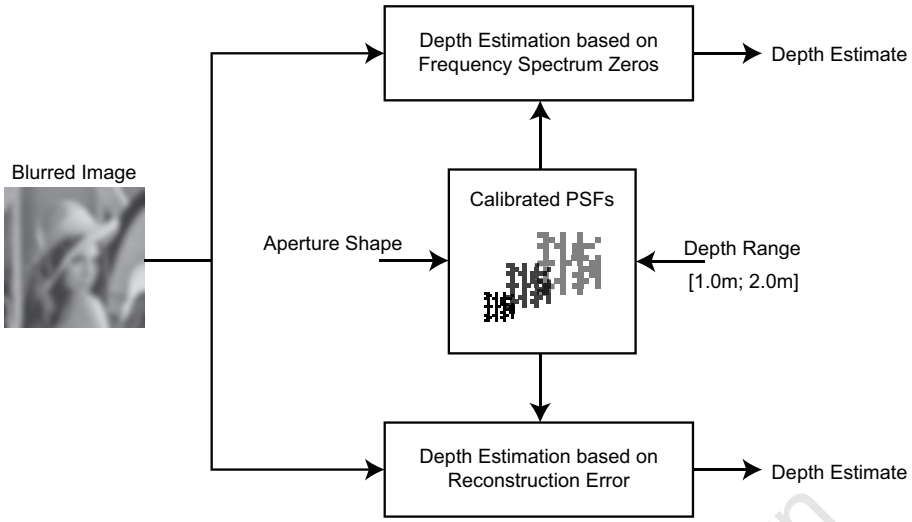



Figure 6.15: Block-diagram showing an overview the depth estimation process.

The two depth estimation algorithms were implemented in MATLAB as functions *estdepth_Pk()* and *estdepth_energy()*, and each requires only the blurred image $y$ and the PSF under consideration $f\_k$. Each PSF in the specified range was tested, and the results from both algorithms were recorded.

## 6.3 Light Field Experiments

Figure 6.16 shows a block-diagram overview of the process that was followed when performing the light field experiments. The first stage consisted of manufacturing the coded apertures required to capture a 4D light field, and constructing the test scene. Multiple photographs were then taken of the test scene, and they were used to generate a digital light field. After the light field had been captured, the following two applications were tested: extracting stereo image pairs for use in a stereo disparity algorithm, and synthesizing virtual images with adjusted optical parameters. Each step will be discussed in more detail in the subsections that follow.

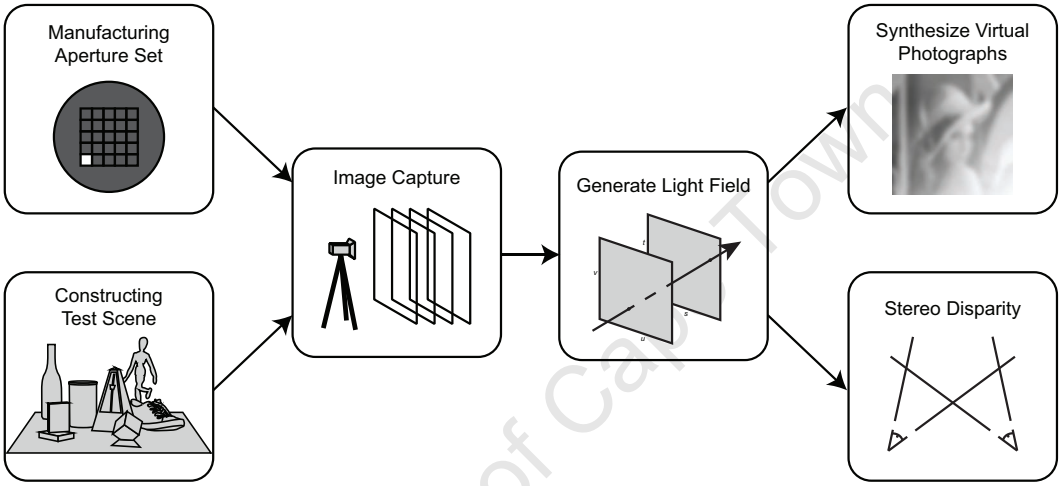



Figure 6.16: Block-diagram showing an overview of the light field experimental procedure.



### 6.3.1 Coded Apertures for Light Field Capture

As discussed in section 4.4, a 4D light field can be captured with a conventional camera by taking multiple photographs using a series of coded apertures. Each coded aperture in the series isolates a single 2D slice of the 4D light field, and the multiple slices can be combined to reconstruct the full signal.

In our experiments we attempted to capture a light field with an angular resolution of $9 \times 9$ (horizontal $\times$ vertical), which requires 81 different $9 \times 9$ binary coded apertures, each with a single transparent block. The apertures were manufactured by cutting the designs out of a cardboard mask, in the same way as the coded apertures for deblurring and depth estimation were manufactured. Figure 6.17 shows four out of the 81 required aperture shapes, and also shows how each aperture was defined by the $(i, j)$ coordinate of its transparent block.

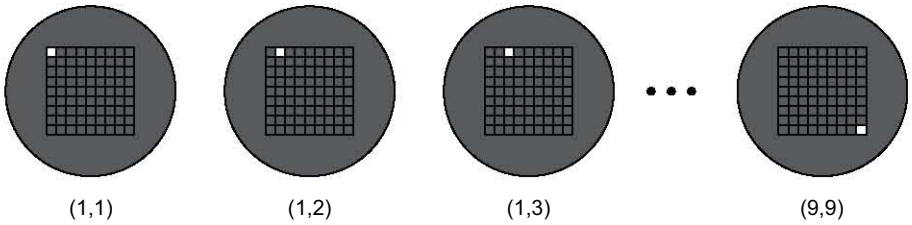



Figure 6.17: Diagram showing four of the 81 required coded aperture shapes used to capture a light field with an angular resolution of $9 \times 9$.

### 6.3.2 Test Scene

In order to thoroughly test the performance of the light field capture, a visually complex test scene was required. As shown in figure 6.18, our test scene consisted of seven unique objects (a wine bottle, a set of playing cards, a cylindrical container, a metronome, a Rubik's cube, a shoe, and a wooden mannequin) placed on a flat surface at different distances from the camera. Within the scene there were a variety of surface textures (diffuse and specular), multiple depth discontinuities and occlusions, as well as opaque, reflective and refractive elements.

The camera was placed on a tripod, and the lens was focused on the central object (the metronome) which was located exactly 1.0m meter away from the camera, while the distances of the other objects ranged from 0.5m to 1.5m. The scene was illuminated using an external flash unit in order to provide constant lighting throughout the 81 required photographs.

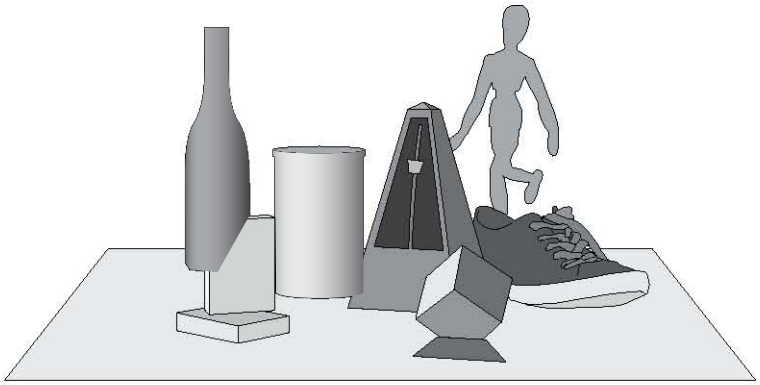


Figure 6.18: Diagram showing the scene layout used for the light field experiments. The scene consisted of a number of interestingly shaped objects placed at different depths on flat surface.

### 6.3.3 Generating the Light Field

Taking care not to alter the camera's position or focus setting, each of the 81 required coded apertures was individually inserted into the lens, and the resulting image was captured and stored on disk.

As discussed in section 4.4, when using coded apertures to capture a light field, the simplest representation is obtained when the $(u, v)$ plane is placed at the aperture plane and the $(s, t)$ plane is placed at the image plane. Under this configuration each binary aperture block defines a point on the $(u, v)$ plane and each pixel on the image sensor defines a point on the $(s, t)$ plane. Therefore each of the 81 captured images represents a single value for $(u, v)$, and the entire light field can be stored as a 2D array of 2D images. This arrangement was shown graphically in figure 2.6.

Following this model, a 4D data-structure was create in MATLAB to store the captured light field, $L(s, t, u, v)$, and the 81 captured images were inserted into this 4D array in the following manner:

$$L(s, t, u, v) = y_{u,v}(s, t), \tag{6.4}$$

where $y_{u,v}$ is the photograph captured with the coded aperture that has its transparent binary block located at $(u, v)$.

### 6.3.4 Stereo Disparity from Light Field

Assuming that the coordinates systems of the $(u, v)$ and $(s, t)$ planes are aligned (which is the case in our experiments), an image obtained from the light field at $(u, v) = (u_0, v_0)$ will always be perfectly aligned to the image obtained at $(u, v) = (u_1, v_0)$ since they lie in the same horizontal line. Therefore all images extracted from a light field for a specific $v$ value can simply be fed into a standard stereo disparity algorithm without requiring any rectification.

The image pair located at $(u, v) = (1, 5)$ and $(u, v) = (9, 5)$ was selected for use in the disparity experiments because it represents the maximum possible horizontal variation over the vertical center of the aperture ($v = 5$). A large horizontal disparity in the image pair increases the accuracy of the stereo calculation, and rays passing through the center of the lens display the least amount of distortion.

There are many freely available stereo disparity implementations to choose from, but the one selected for this experiment was developed by Lankton [17], and is based on a research paper by Klaus et al. [14]. The algorithm is implemented in MATLAB, and only has one tunable parameter, *win_size*, which specifies the size of the smoothing window. The algo-

rithm produces a pixel disparity map as its output, which can be converted to a depth map if the intrinsic parameters of the camera are known.

### 6.3.5 Synthesizing Virtual Photographs

As explained in section 4.4, a virtual photograph can be synthesized from a light field by placing a virtual image plane into the standard two-plane model, and integrating over the angular dimensions of the light field. The location of the focal plane in the virtual photograph can be adjusted by moving the virtual image plane away from or closer to the aperture plane, and the aperture shape can be adjusted by only integrating over a subset of angular dimensions.

The methods described in section 4.4 were implemented in MATLAB as the function, *lfsnapshot()*. The function takes the specified aperture shape, $f_k$, and the distance between the virtual image plane and the aperture plane, $\alpha$, as inputs, and outputs a 2D virtual photograph. The distance between the image plane and the aperture plane is normalized to 1.0, and therefore $\alpha < 1$ represents a virtual focal plane further away then the original focal plane, while $\alpha > 1$ represents a virtual focal plane closer than the original.

Three different virtual aperture shapes were tested with the virtual image synthesis algorithm, namely: a centered diamond, a diagonal line, and a fully open aperture. These three apertures are shown in figure 6.19 (a), (b), and (c) respectively. The refocussing was then tested by generating virtual photographs with $\alpha$ values ranging from 0.98 to 1.02 in 0.001 increments.

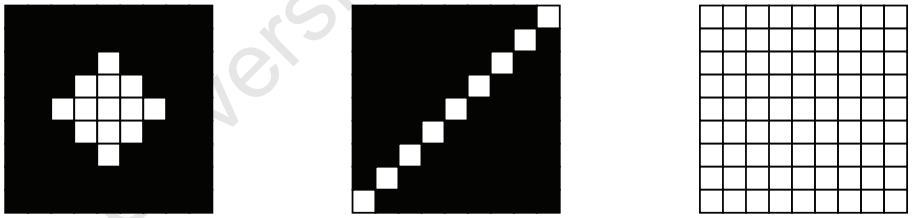



Figure 6.19: Diagram showing the three virtual aperture shapes that were used to test the virtual image synthesis algorithm. (a) A centered diamond shape, (b) a diagonal line, and (c) a fully open aperture.

## 6.4 Coded Exposure Experiments

Figure 6.20 shows a block-diagram that provides an overview of the coded exposure experimental procedure. The first step was to select the coded exposure patterns that would be used in the experiments, and to construct the test scenes. The selected exposure patterns were then used to capture photographs of the moving test scenes in order to create images with coded motion blur. Next, using knowledge of the exposure pattern and scene movement, the PSF for each motion blurred image was estimated. Lastly, an attempt to remove the motion blur was made by deblurring the captured images using the estimated PSFs. The rest of this section is devoted to discussing each step of the experimental procedure in more detail.

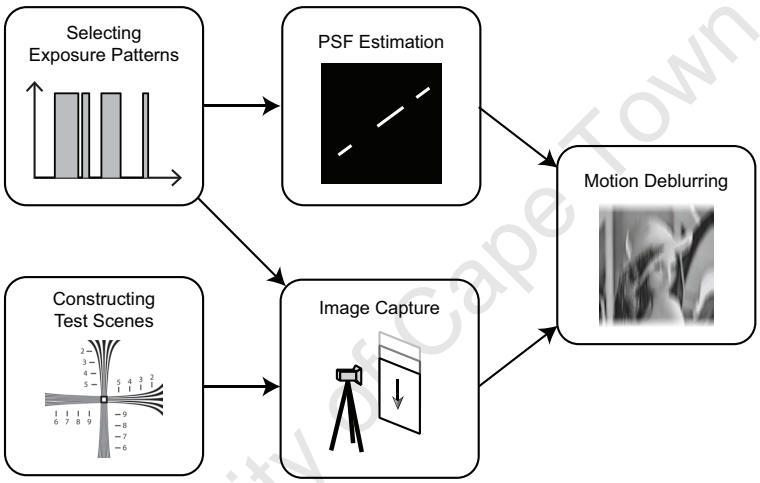



Figure 6.20: Block-diagram showing an overview of the coded exposure experimental procedure.



### 6.4.1 Selecting Exposure Patterns

In chapter 5, the theoretical methods for finding optimal motion deblurring patterns were presented, and in this section these methods are practically implemented and the resulting exposure patterns are presented and discussed. All the patterns used in the coded exposure experiments were of length $N = 52$, since this was the length that was suggested by Raskar et al. [29].

#### Optimal Exposure Patterns for Motion Deblurring

In section 5.2, two fitness functions, $R_{raskar}$ and $R_{zhou}$, were developed, and are defined in equations 5.2 and 5.2 respectively. $R_{raskar}$ requires only the exposure pattern under inves-

tigation, $f_k$, as input, since the output of the function is based solely only the shape of the exposure pattern's Fourier transform. On the other hand, $R_{zhou}$ requires three parameters: the pattern under consideration $f_k$, the standard deviation of the noise in the observed blurred image $\sigma$, and an approximation for the power spectrum of natural images $A(\omega)$. The values used for $\sigma$ and $A(\omega)$ were identical to the values used in section 6.2 for selection of optimized deblurring apertures.

To search for the optimal exposure patterns, the fitness functions $R_{raskar}$ and $R_{zhou}$ were used in conjunction with the genetic algorithm used in section 6.2. Figures 6.21 and 6.22 show the how the genetic algorithm converges to a local minimum for $R_{raskar}$ after about 150 generations, and for $R_{zhou}$ after about 30 generations.

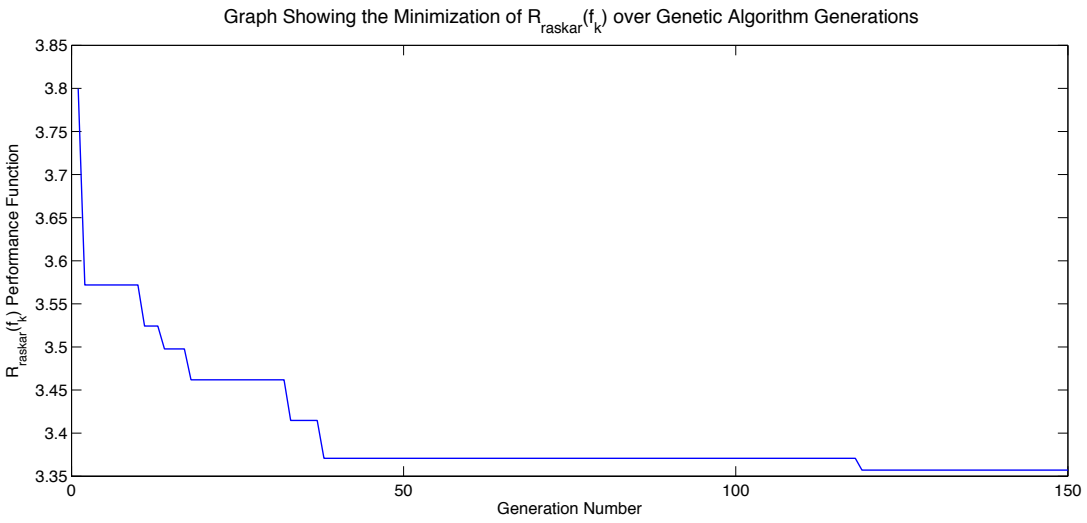



Figure 6.21: Graph showing the minimum $R_{raskar}(f_k)$ value for each generation of the genetic algorithm. The graph shows that the algorithm converges to a local minimum between 100 and 150 generations.

### Other Exposure Patterns for Comparison

In addition to the two optimized exposure patterns that have been selected for motion deblurring, the coded exposure experiments were also performed with a conventional pulse exposure and the coded exposure pattern developed by Raskar et al. [29]. The two additional patterns were included in order to compare their performances with that of the optimized patterns, and thereby judge the effectiveness of the optimization methods. For fairness the conventional pulse exposure pattern was scaled to have the same total exposure time as the coded patterns. The two optimized patterns and the two additional patterns are defined in

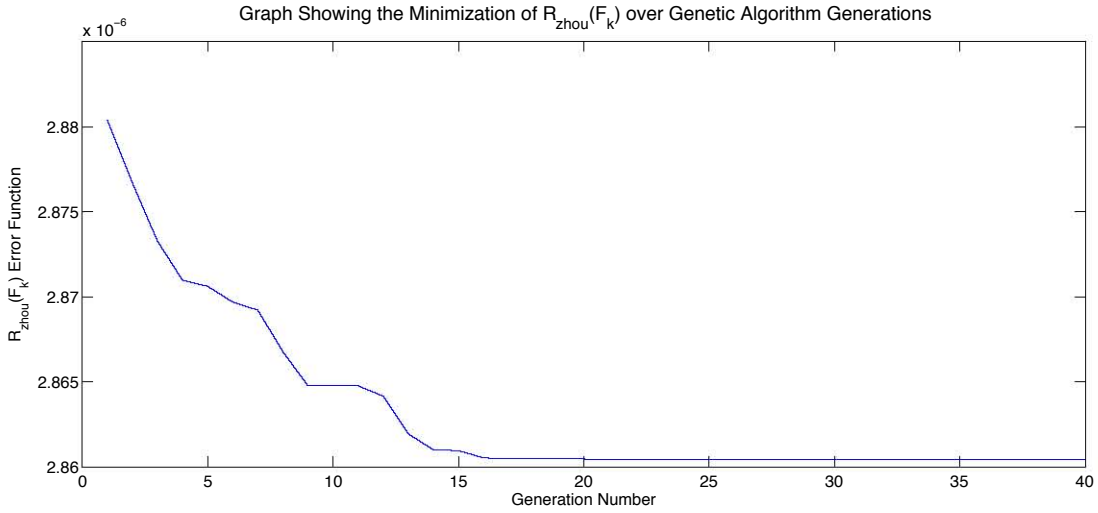



Figure 6.22: Graph showing the minimum $R_{zhou}(F_k)$ value for each generation of the genetic algorithm. The graph shows that the algorithm converges to a local minimum after about 30 generations.

the table below, and they are also shown graphically over time in figure 6.23. Although the exposure patterns are represented by binary strings, hexadecimal notation is used in the table for clarity.

| | |
|---|---|
| **Raskar** | 0xA1C1433DD7267 |
| $\mathbf{R_{raskar}(f_k)}$ | 0xAB6361069AE43 |
| $\mathbf{R_{zhou}(F_k)}$ | 0xFFF8000FC01FF |
| **Conventional** | 0xFFFFFFC000000 |

### 6.4.2 Test Scenes

The coded exposure experiments used the same resolution chart scene and human face scene that were used in the coded aperture experiments. In all the experiments the camera was focused at 1.0m and was located 1.0m away from the scenes. In order to produce motion blur in the observed images, a relative motion has to exist between the scene and the camera. Therefore, during the capture interval the resolution chart was moved vertically upwards by hand, and the human test subject moved their face vertically downwards by slowly bending their legs. The goal was to achieve an approximately constant speed despite the crude methods used. The total exposure time for the test scenes was 0.5s and on average the scene objects travelled approximately 20cm during their exposure, thereby producing a significant amount of motion blur. The layout of the test scenes are shown in figure 6.24.

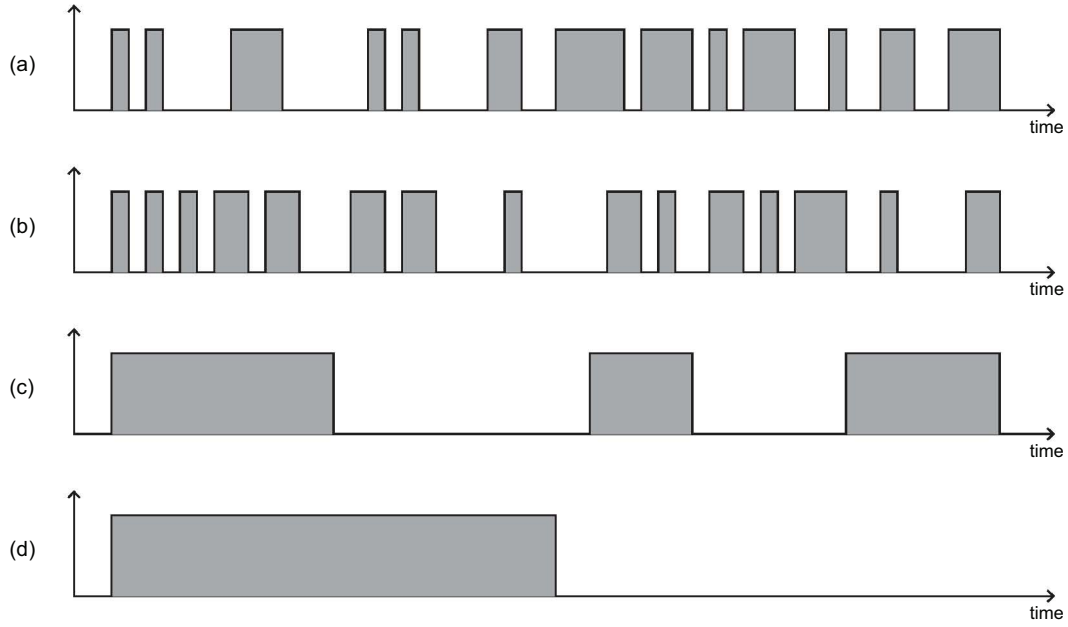



Figure 6.23: Diagram showing the coded exposure patterns that were used in the experiments. (a) The pattern used by Raskar et al. in their paper [29], (b) the pattern optimized using the $R_{raskar}(f_k)$ performance metric, (c) the pattern optimized using the $R_{zhou}(F_k)$ performance metric, and (d) a conventional pulse exposure.

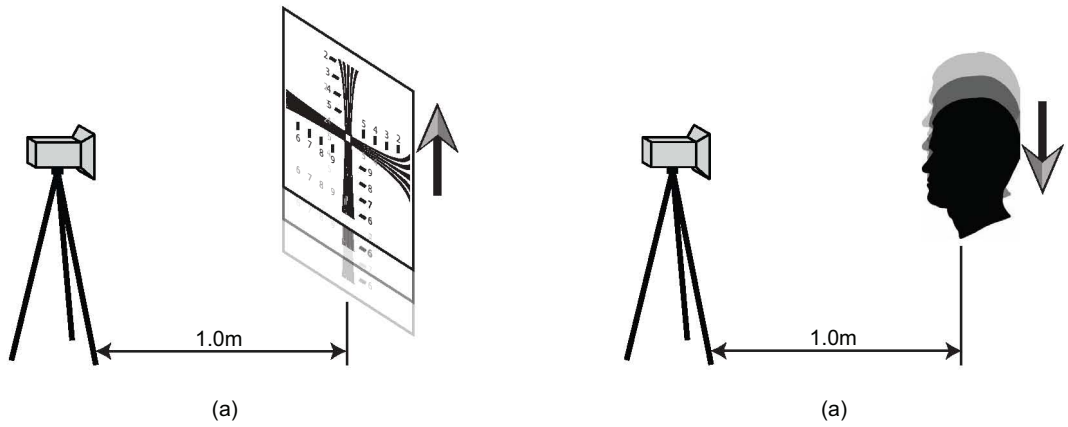



Figure 6.24: Diagram showing the two test scenes that were used in the coded exposure experiments. (a) A planar resolution chart (extract from the ISO 12233 resolution chart) moving upward at a constant speed, and (b) a human face moving downward at a constant speed.

### 6.4.3 Motion Deblurring

#### Estimating the PSF

Before the observed images can be deblurred, the PSF for each photograph has to be estimated. As discussed in section 5.2, the shape of the PSF is determined by the exposure

pattern and the direction of relative motion, while the scale of the PSF can be calculated from the extent to which objects in the observed images have become elongated. Since the motion in the test scenes was always vertical, the shape of the PSFs was always assumed to be a vertical line modulated by the appropriate binary exposure pattern. The length of each PSF was calculated manually by subtracting the vertical pixel length of a known sharp object from the length of its blurred image in the observed photographs. Figure 6.25 shows four example PSFs that were estimated using this procedure.

### Deconvolution

Once the PSF for each observed image has been estimated, all that remains is to input the observed image and the estimated PSF into a deconvolution algorithm. For the sake of consistency the same deconvolution algorithm that was used in the defocus deblurring experiments (section 6.2) was used again for the motion deblurring experiments.

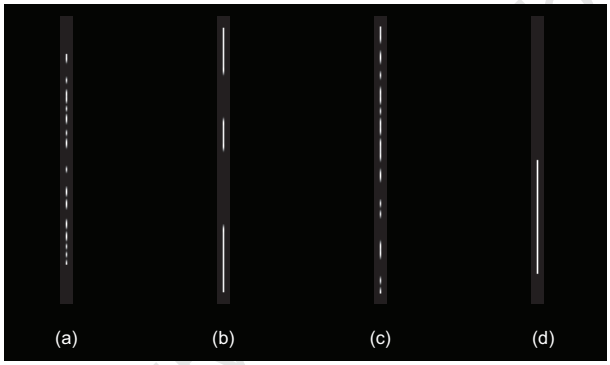



Figure 6.25: Diagram showing example PSFs for vertical motion using coded exposure patterns: (a) optimized using $R_{raskar}$, (b) optimized using $R_{zhou}$, (c) taken from the paper by Raskar et al., and (d) a conventional pulse.

# Chapter 7

# Results

## 7.1 Coded Aperture Results

### 7.1.1 Aperture Calibration

Figure 7.1 (page 92) shows the results obtained from the aperture calibration procedure detailed in section 6.2. Five apertures were calibrated: the refocus-optimized aperture, the depth-optimized aperture, a conventional circular aperture, the aperture chosen by Zhou et al. [39], and the aperture chosen by Levin et al. [18]. In order to make the shapes of the PSFs easily visible, the results in the diagram have been normalized so that their maximum values are equal. However, when actually used for deblurring and depth estimation, the PSFs were normalized so that the sum of their intensities are equal.

The results clearly show how the shapes of the defocus PSFs are equal to the shapes of the physical apertures (shown in figures 6.6 (b), 6.8 (e) and 6.9), while the scale of the PSFs are determined by the distance from the focal plane (located at 1.0m in our experiments). It is also important to note how the rate of change of PSF scale decreases as the distance from the focal plane increases. This suggests that it becomes more difficult to determine depth from PSF scale as the distance from the focal plane increases.

Due to the small scale of PSFs that are located close to the focal plane, the structure of the aperture shape only becomes visibly apparent for distances of 1.2m and greater. This suggests that the advantages associated with using coded apertures become less apparent as scene objects approach the focal plane, and the shape of the PSF tends to a circular point.

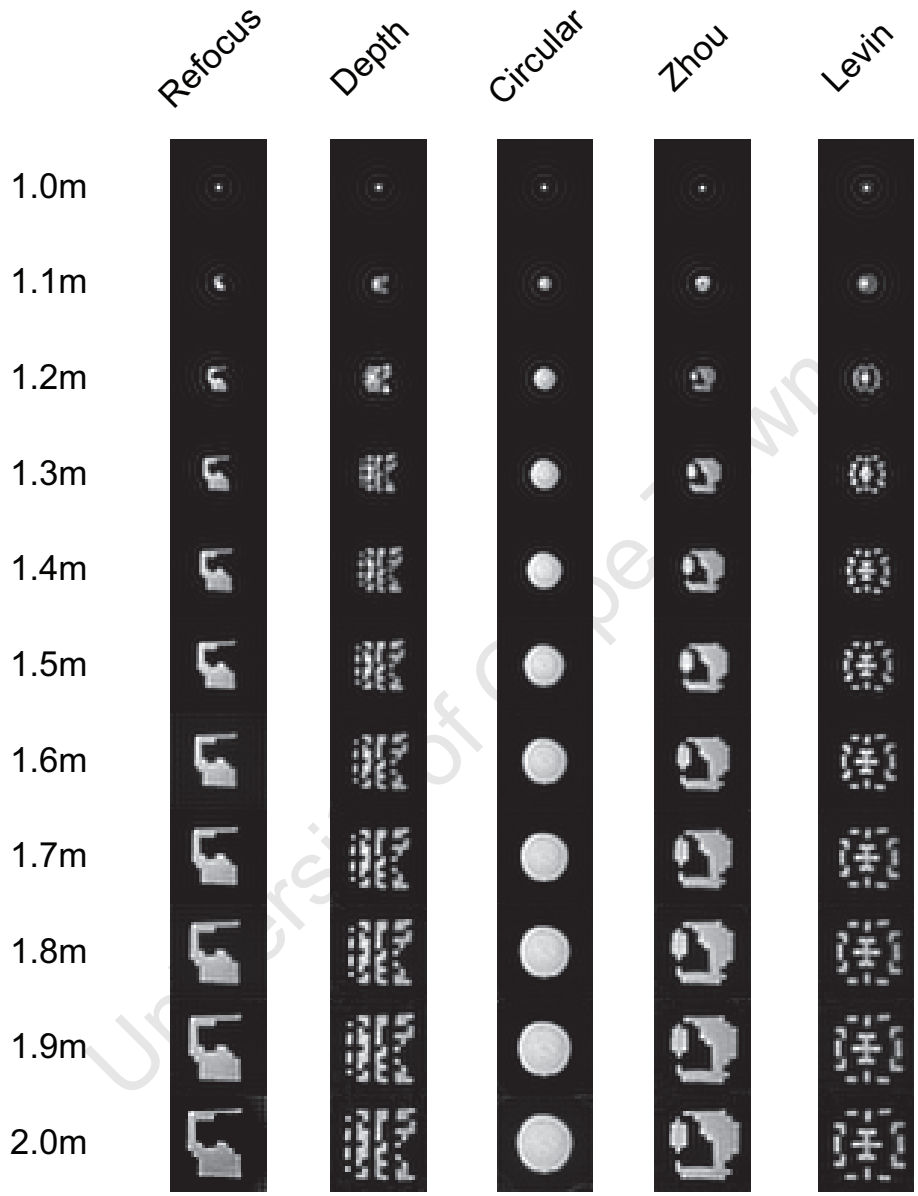



Figure 7.1: Diagram showing the results obtained from aperture calibration. The PSFs for each of the five aperture shapes used are shown for depths ranging from 1.0m (the focused distance) to 2.0m in 10cm increments.

### 7.1.2 Defocus Deblurring

The results of defocus deblurring using the calibrated PSFs from the previous section are shown in figures 7.2 – 7.9 (pages 96 – 103). Due to space restrictions, not all of the defocus deblurring results could be included in this report, and the results that have been included may not be of sufficient size to see all the effects that will be discussed. Therefore it is highly recommended that the reader see the full-resolution results included electronically in appendix A.4.

**Deblurring at 1.2m**

Figures 7.2 and 7.3 show the deblurring results for the resolution chart scene and the human face scene, both located 1.2m away from the camera. Since this is only 20cm away from the focal plane, the PSFs are small and the observed images are only slightly blurred. The blurring is only really noticeable at visual edges such as the lines in the resolution chart and the fine detail in the face. Importantly, the blurring is so slight that the specular highlights in the face's eyes are well defined, and in fact hit the upper exposure limit of the camera's sensor.

At 1.2m, the observed images captured using the circular aperture have lost definition in the finest lines of the resolution chart, and the fine details of the face. Also, a zero frequency band is quite clearly visible in the horizontal lines of the resolution chart between the 6 and 7 markers. All line structure is lost in these bands due to the zeros in the PSF underlying frequency spectrum. The circular aperture's deblurred images successfully recover most of the hard edges, but they also include significant ringing in high-contrast areas. Also, the non-linearity caused by the specular highlight hitting the sensor's upper limit has introduced significant artifacts.

At 1.2m, the observed images captured using the depth-optimized aperture and Levin's aperture also contain visible zero-bands in the resolution chart. This is not surprising since well-defined zeros in the frequency domain was one of the criteria used in their optimization. These apertures performed better at deblurring than the circular aperture, but significant ringing is still present and information located at their zero bands is not recovered at all.

At 1.2m, the observed images captured using the refocus-optimized aperture and Zhou's aperture have also lost their hard edges, but the blur in these images seems to be more structured than the smooth blur caused by the circular aperture, and no zero bands can be seen in the resolution chart. This is to be expected since these apertures were optimized to have high minima in their frequency domains. The deblurred results for these apertures clearly outperform the others. All hard edges are recovered and even the finest lines in

the resolution chart can be distinguished. There is still a slight amount of ringing at high contrast edges, and the same nonlinear artifacts are present at the specular highlights.

**Deblurring at 1.6m**

Figures 7.4 and 7.5 also show the deblurring results for the resolution chart scene and the human face scene, but this time both scenes are located 1.6m away from the camera. Being further away from the camera's focal plane, the blurring in these images is predictably more severe. The text in the resolution chart is now unreadable, and no detail is visible in any of the features of the human face. Also, the larger PSF scale at this distance has dispersed the intensity of the specular highlights across multiple pixels, thereby bringing them back into the sensor's linear range.

At 1.6m, the circular aperture's observed images have lost a significant amount of high frequency detail, making all but the coarsest lines in the resolution chart indistinguishable. The zero bands in the horizontal lines that were present at 1.2m are now more numerous and more pronounced. In the deblurred images the text is has been recovered enough to be readable, and the contrast in the coarsest details has improved. However, details at the zero bands are completely unrecovered and significant ringing is present. Interestingly, the non-linear artifacts that were present at 1.2m are no longer visible.

At 1.6m, the observed images obtained using the depth-optimized aperture and Levin's aperture again contain multiple zero-bands visible in the horizontal lines of the resolution chart. It is noteworthy that these zero bands are not spaced as equally as the zero bands in the circular aperture's image, and this confirms the hypothesis that the bands are located at the PSF's frequency spectrum zeros. The result of deblurring with these apertures is again better than that of the circular aperture, since the text is recovered quite well, and the ringing that is present is visibly less pronounced. Unfortunately, detail at the numerous zero bands is unrecovered.

At 1.6m, the observed images obtained using the refocus-optimized aperture and Zhou's aperture show an interestingly structured blur. Instead of losing definition between lines in the resolution chart, the defocus blur seems to distort the lines without degrading their edges. While some areas of the resolution chart are better preserved than others, there are no areas in which all definition is lost. This intuitive analysis is supported by the high performance of the deblurring results. The text is completely recovered, as well as all but the finest details in the chart and the face. Only slight ringing is introduced, and it seems to be more affected by the depth-discontinuities in the 3D face than by the planar edges in the resolution chart.

**Deblurring at 2.0m**

Figures 7.6 and 7.7 again show the deblurring results for the resolution chart scene and the human face scene, but now located 2.0m away from the camera. This was the maximum distance used in the deblurring experiments, and therefore these figures show the maximum amount of defocus blur. The text of the resolution chart now appears as nothing more than dark smudges, and while the overall shape of the chart can still be seen, none of the individual lines are discernible. In the images of the face, only the largest features can be identified, and the specular highlights in the eyes are no longer visible.

At 2.0m, the circular aperture's observed images contain severe and smooth blurring effects. None of the lines in the resolution chart are distinguishable, and even moderately large facial features (e.g. the eyes, nose, and ears) are badly defined. In the deblurred images, only some of the textual characters can be speculated, and all but the coarsest details remain unrecovered. These images also show the most severe ringing effects in the entire experiment, and they are located at both contrast edges and depth edges.

In the case of the depth-optimized aperture and Levin's aperture, the deblurring results are far better than that of the circular aperture. The text is recovered enough to be identified, but the characters do show some distortion. Only the medium-to-coarse lines in the resolution chart can be distinguished, and the lines located at the many zero bands are unrecovered. Moderate ringing is also present in the deblurred images, but to a lesser extent than with the circular aperture.

Finally, in the observed images obtained using the refocus-optimized aperture and Zhou's aperture, while the severe blurring allows for no distinguishable lines in the resolution chart, the blur still seems to contain a visible amount of high frequency structure. Their deblurred images are clearly superior to the other apertures discussed, as only the finest details have been lost in the face, and the text in the resolution chart has been recovered perfectly. Again, only a small amount of ringing is present, and it seems to be much more prominent in the image of the face than in the image of the resolution chart. This suggests that the ringing might be caused by the non-planar depth of the human face, rather than by a loss of frequency information.

Figures 7.8 and 7.9 show zoomed versions of the deblurred results at 2.0m for the circular, depth-optimized and refocus-optimized apertures. The superiority of the refocus-optimized aperture is most noticeable at this maximum level of blurring, and this may be due to the fact that miscalibration of the PSF is more likely at distances closer to the focal plane.

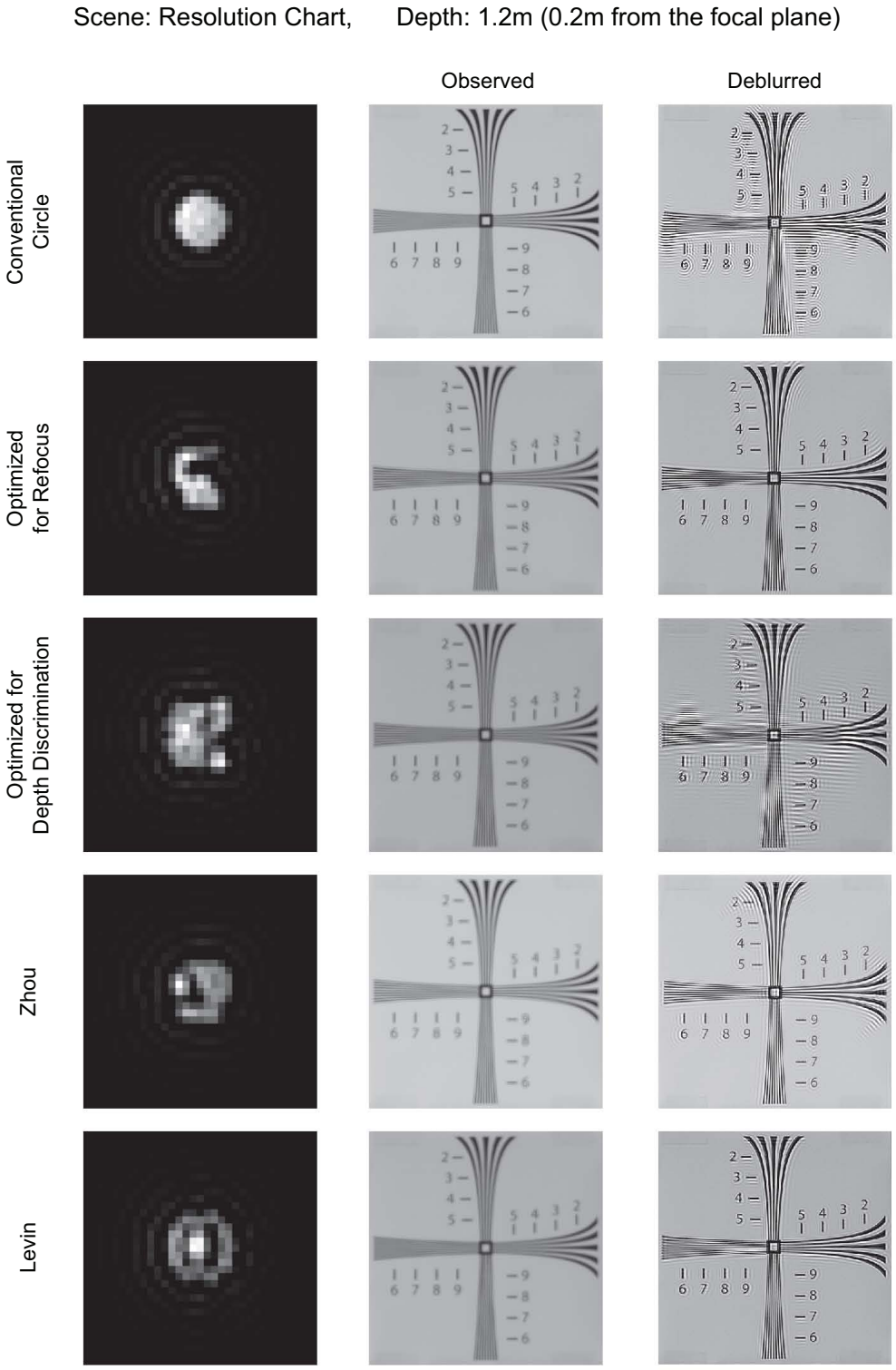



Figure 7.2: Results of the deblurring experiment using the resolution chart scene at a distance of 1.2m.

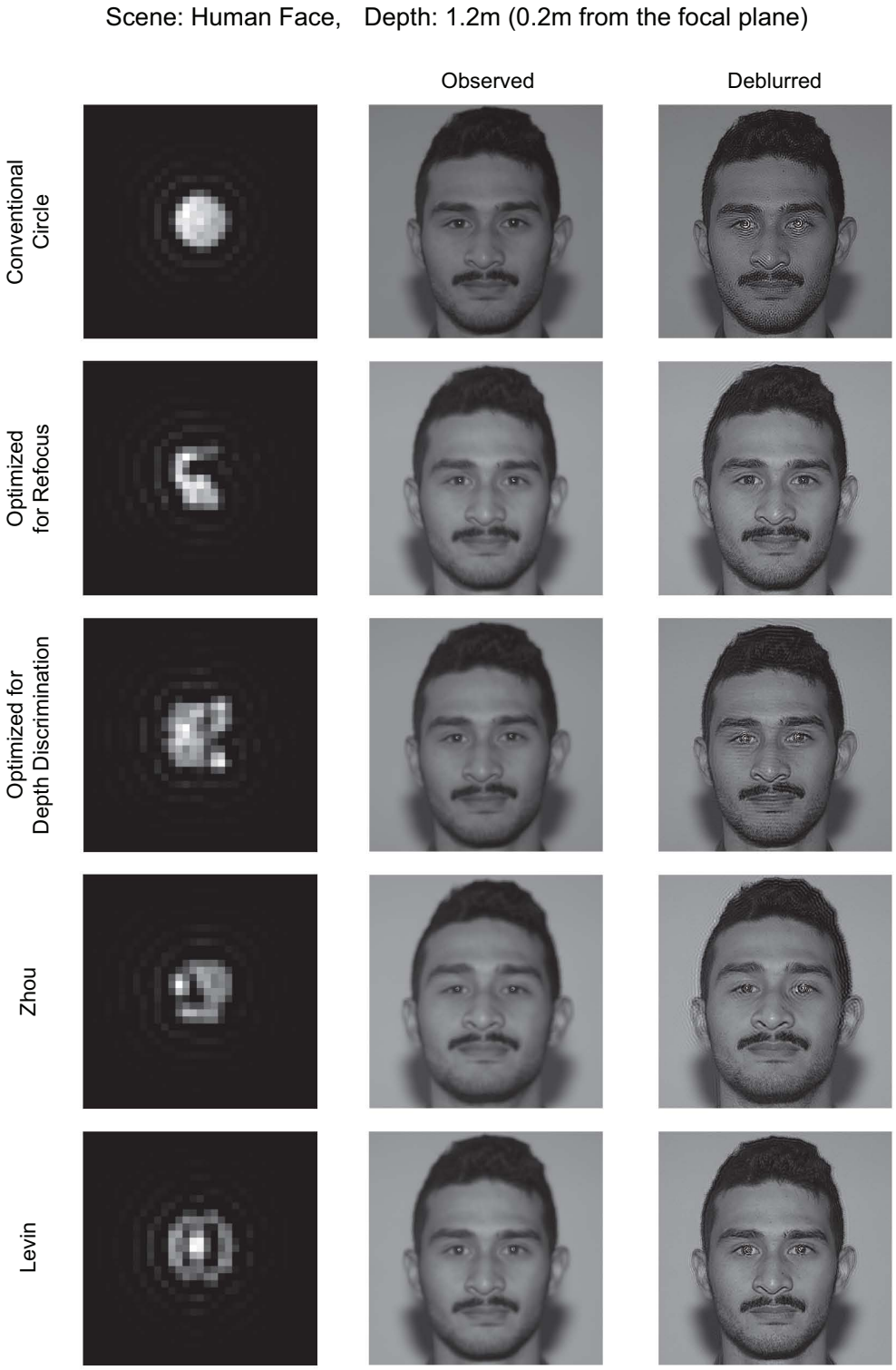



Figure 7.3: Results of the deblurring experiment using the human face scene at a distance of 1.2m.

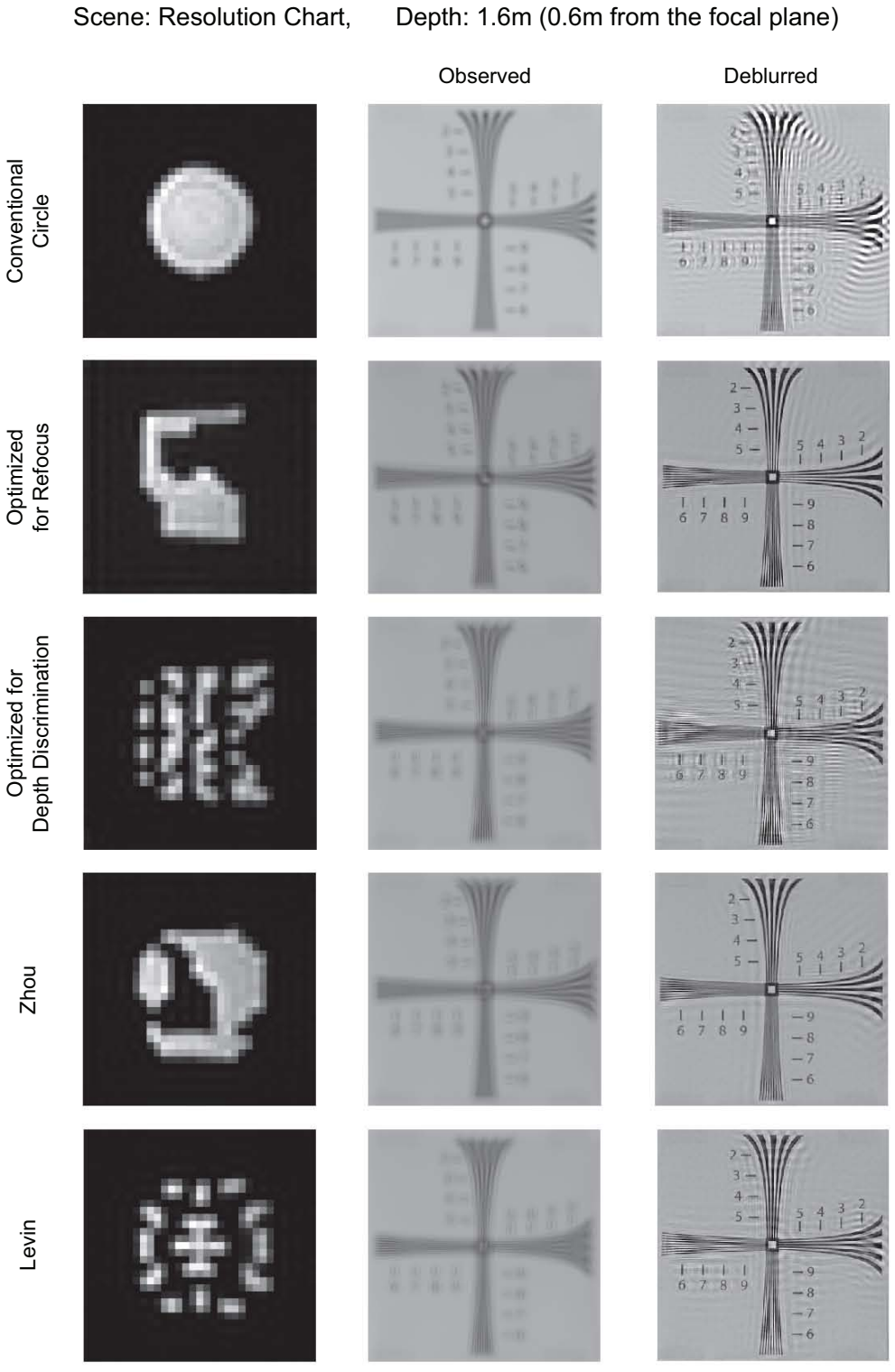



Figure 7.4: Results of the deblurring experiment using the resolution chart scene at a distance of 1.6m.

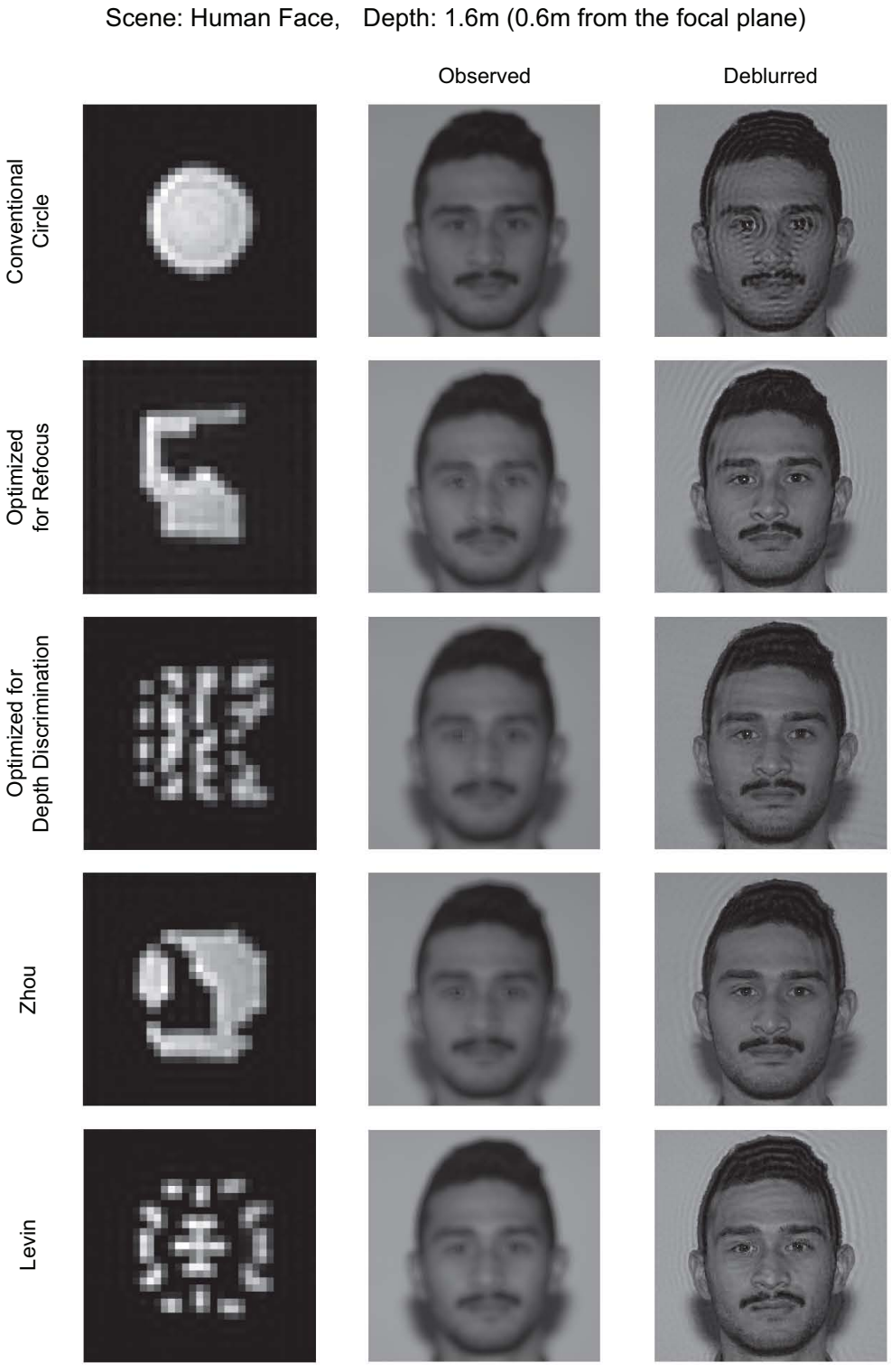



Figure 7.5: Results of the deblurring experiment using the human face scene at a distance of 1.6m.

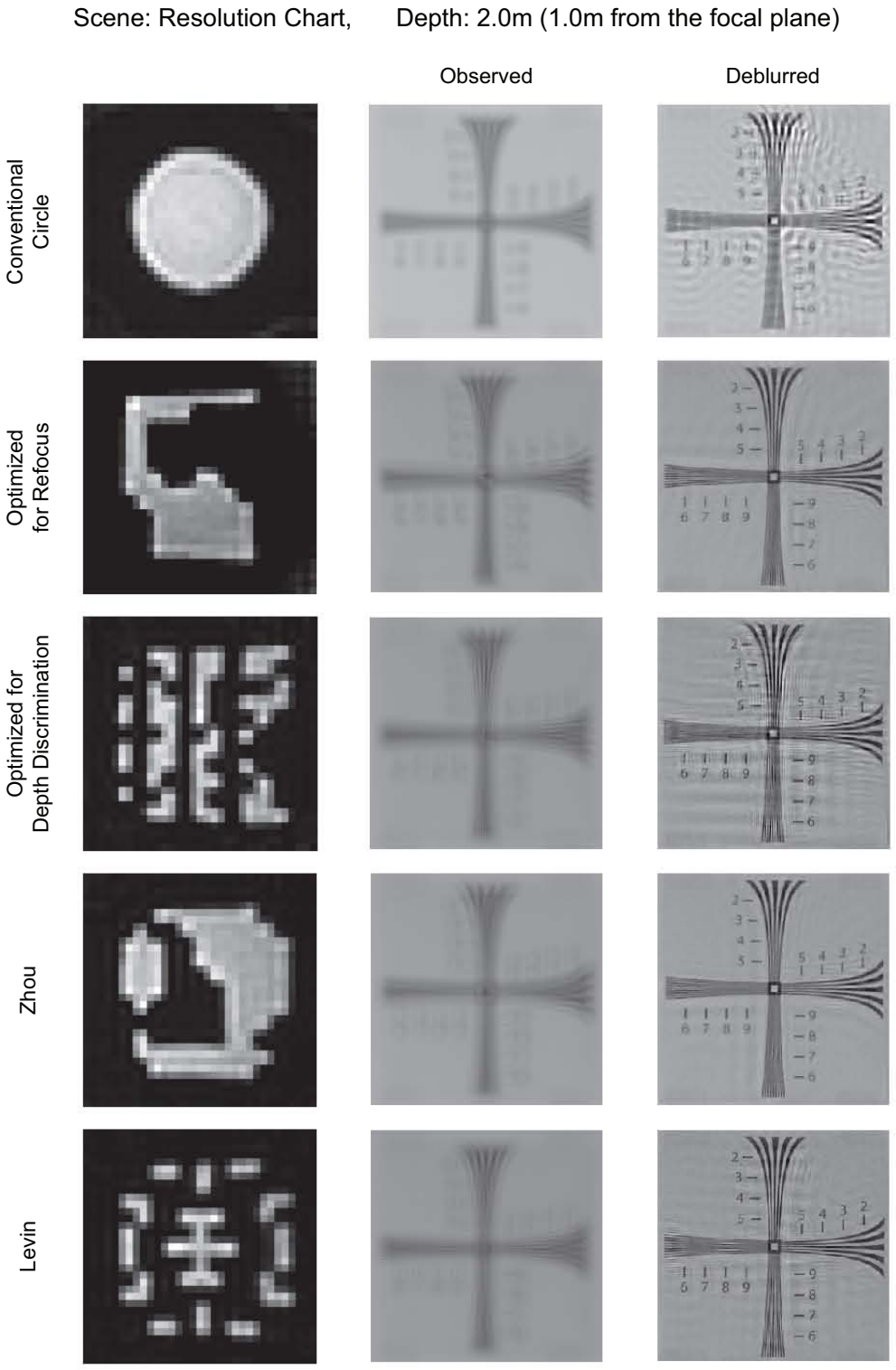



Figure 7.6: Results of the deblurring experiment using the resolution chart scene at a distance of 2.0m.

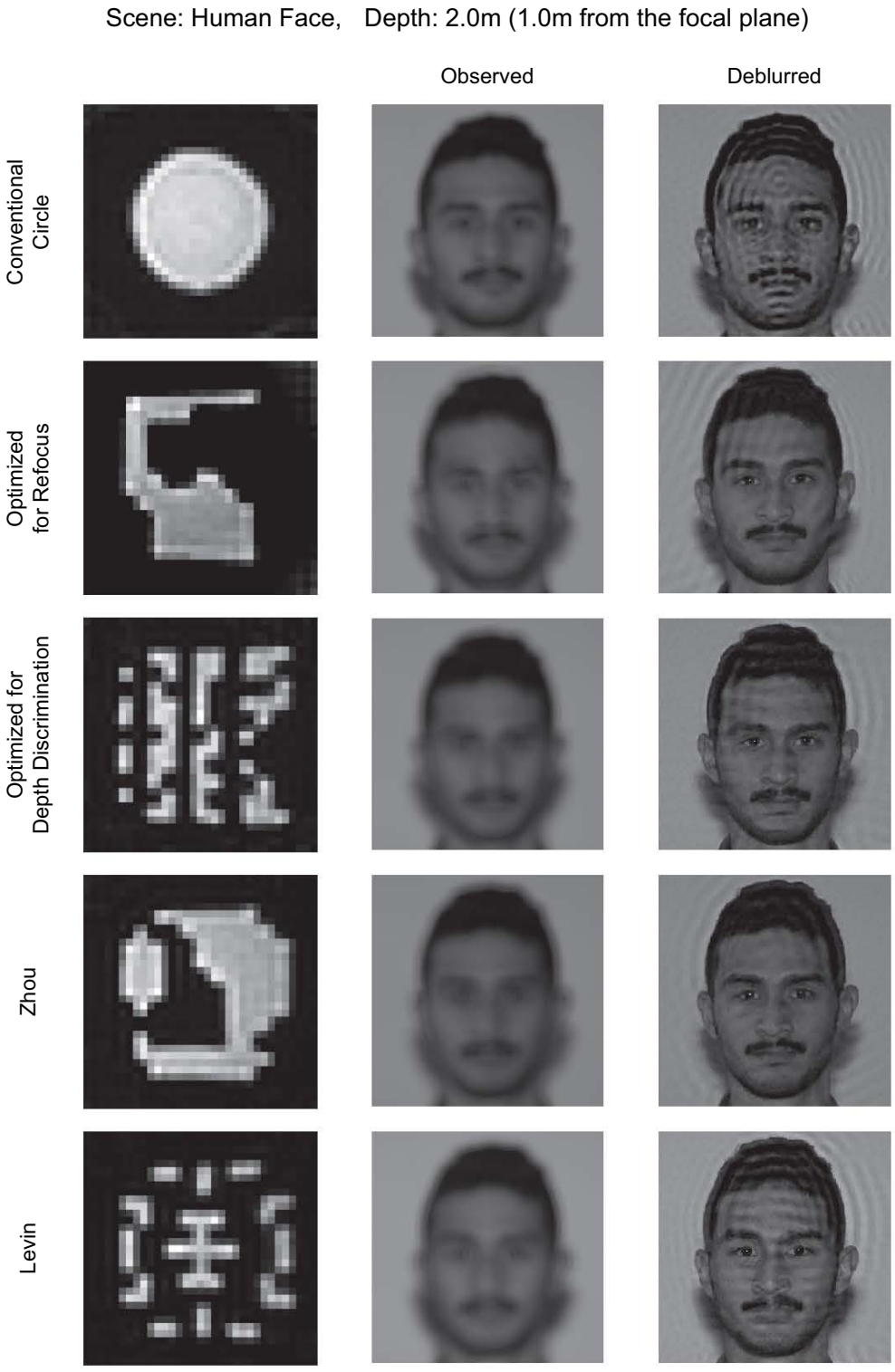



Figure 7.7: Results of the deblurring experiment using the human face scene at a distance of 2m0.

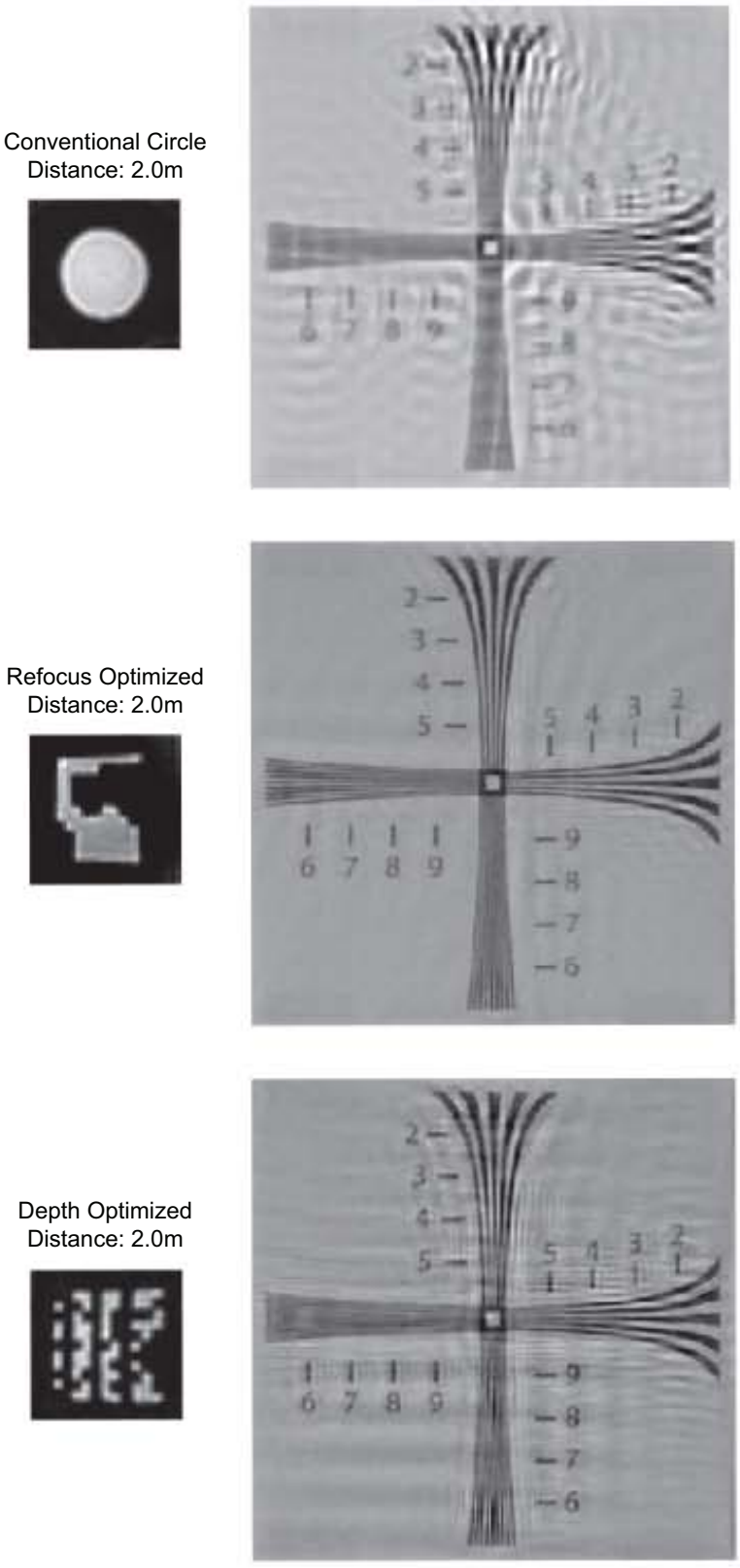



Figure 7.8: Diagram comparing the resolution chart deblurring results obtained using a conventional aperture, a coded aperture optimized for refocusing, and a coded aperture optimized for depth discrimination.

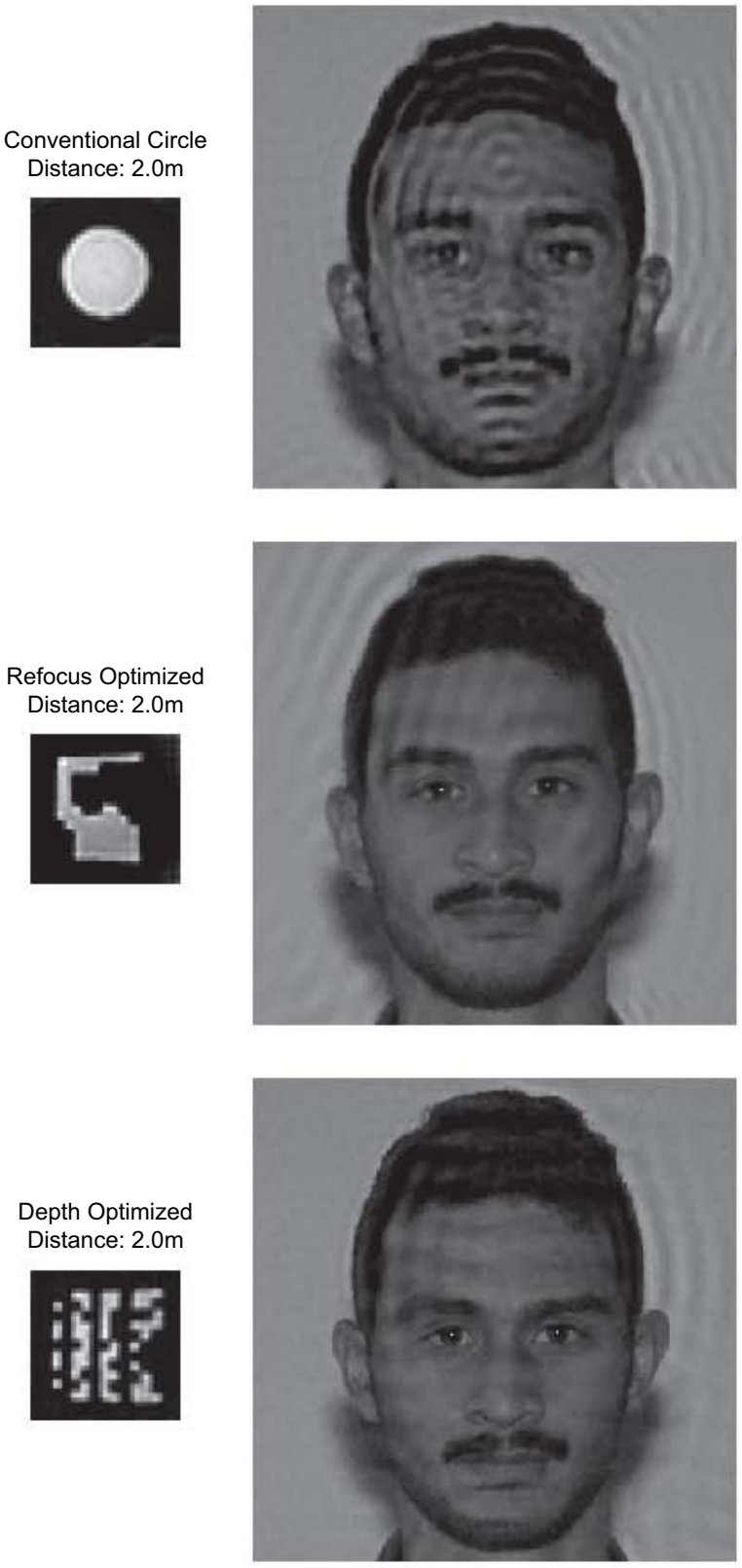



Figure 7.9: Diagram comparing the human face deblurring results obtained using a conventional aperture, a coded aperture optimized for refocusing, and a coded aperture optimized for depth discrimination.

### 7.1.3 Depth Estimation

Two different methods for estimating depth from coded aperture defocus were described in section 4.3. The first uses the probability function $P_k(y)$ which must be calculated for an observed image, $y$, over all possible PSF scales, $k$, and the scale that maximizes the functions value is used to determine the scene depth. The second method calculates the value of the function $E_k(y)$ (which is approximated by the error introduced when a blurry image is deblurred using the incorrect depth) for each possible PSF scale, $k$, and then selects the scale that produces the minimum value. Both methods were tested in the experiments, and the results are shown in figures 7.10 – 7.15 (pages 106 – 111). While 11 scene depths (1.0m to 2.0m in 10cm increments) were used in the depth experiments, due to space constraints the above figures only show the results for six scene depths (1.0m – 2.0m in 20cm increments). This is done since the general trend can still be seen from the six depths, and the full set of results can be found electronically in appendix A.4.

In both the methods described in section 4.3, it was suggested that the raw values of $P_k(y)$ and $E_k(y)$ be multiplied by a weighting vector before the maximum or minimum value is chosen. The weighting vector is required in order to normalize the intrinsic weighting associated with each PSF scale (especially at distances near the focal plane), and it needs to be calculated by running the experiments on multiple training sets (with known depths), and optimizing the weights to minimize the classification error. This vital extra step was not performed in our experiments, so the depth of each image cannot be deduced by taking the simple minimum or maximum value. Instead, the trend of each function over the possible depth values is analyzed for local minima and maxima.

Figures 7.10 and 7.11 show the values of $P_k(y)$ (left column) and $E_k(y)$ (right column) for the photographs taken with a conventional circular aperture of the resolution chart scene and complex planar scene respectively. At a distance of 1.0m, both functions clearly indicate that the most likely distance is 1.0m. However, this represents a trivial case since the camera's focal plane was set at 1.0m, thereby reducing the depth estimation problem to one of detecting whether the input image is in focus or not. However, the results produced at 1.2m are slightly more insightful because extrema are located at 1.1m in both functions. Although these extrema are located 10cm away from the true depth, their presence indicates that the two methods are able to detect that the input images are not perfectly sharp, but are blurred with a small-scale PSF. Unfortunately the performance of the depth estimation methods decrease as the depth of the input image increases. For input images at depths of 1.4m to 2.0m, the values of $P_f(k)$ decrease monotonically (without any local maximum) over the depth axis, and the values of $E_k(y)$ increase similarly, except with a false local minimum fixed at 1.6m. Judging from these plots it is quite clear that a circular aperture does not produce PSF scales that are sufficiently identifiable for depth estimation. Even if

a optimized weighting vector were to be used in the experiment, the raw function values are too similar across multiple image depths to produce usable results.

Figures 7.12 and 7.13 show the results of the same experiments as above, except now using the depth-optimized coded aperture instead of the conventional aperture. Except for images at 1.2m, the shape of the $P_k(y)$ plots show local maxima located close ($\pm$10cm) to the true depths. A possible explanation for why an accurate result was not obtained for a depth of 1.2m is that the PSF scale at this depth is just too small to properly define its carefully engineered shape. The results obtained from the $P_k(y)$ function are accurate for both the resolution chart scene and the complex planar scene, and this suggests that this method will work regardless of whether the scene is visually complex or not. Unfortunately the results obtained using the $E_k(y)$ function are much less successful. Other than the trivial 1.0m case, the function's local minima do not occur at the scene's true depth, and instead seem to be fixed at certain positions. The aperture used in this experiment was optimized using a derivative of the $P_k(y)$ function and this is most likely to be reason why the depth estimation method using $P_k(y)$ performs so much better than the method based on $E_k(y)$.

Finally, figures 7.14 and 7.15 show the depth estimation results for the resolution chart scene and the complex planar scene, using the aperture developed by Levin et al. [18]. Using this aperture, both the $P_k(y)$ function and the $E_k(y)$ function produce easily identifiable local extrema located near the true scene depths. As was the case for the previous aperture shape, scenes at 1.2m are not correctly classified, and the same explanation is postulated. From the results of these experiments, Levin et al.'s aperture seems to produce the best results for both depth estimation methods. This not surprising since the methods used here were also suggested in their paper. The only significant difference between Levin et al.'s aperture optimization and our own is that Levin et al. explicitly added the constraint of aperture symmetry, and perhaps this is the reason why their aperture performs better than ours when using the $E_k(y)$ function.

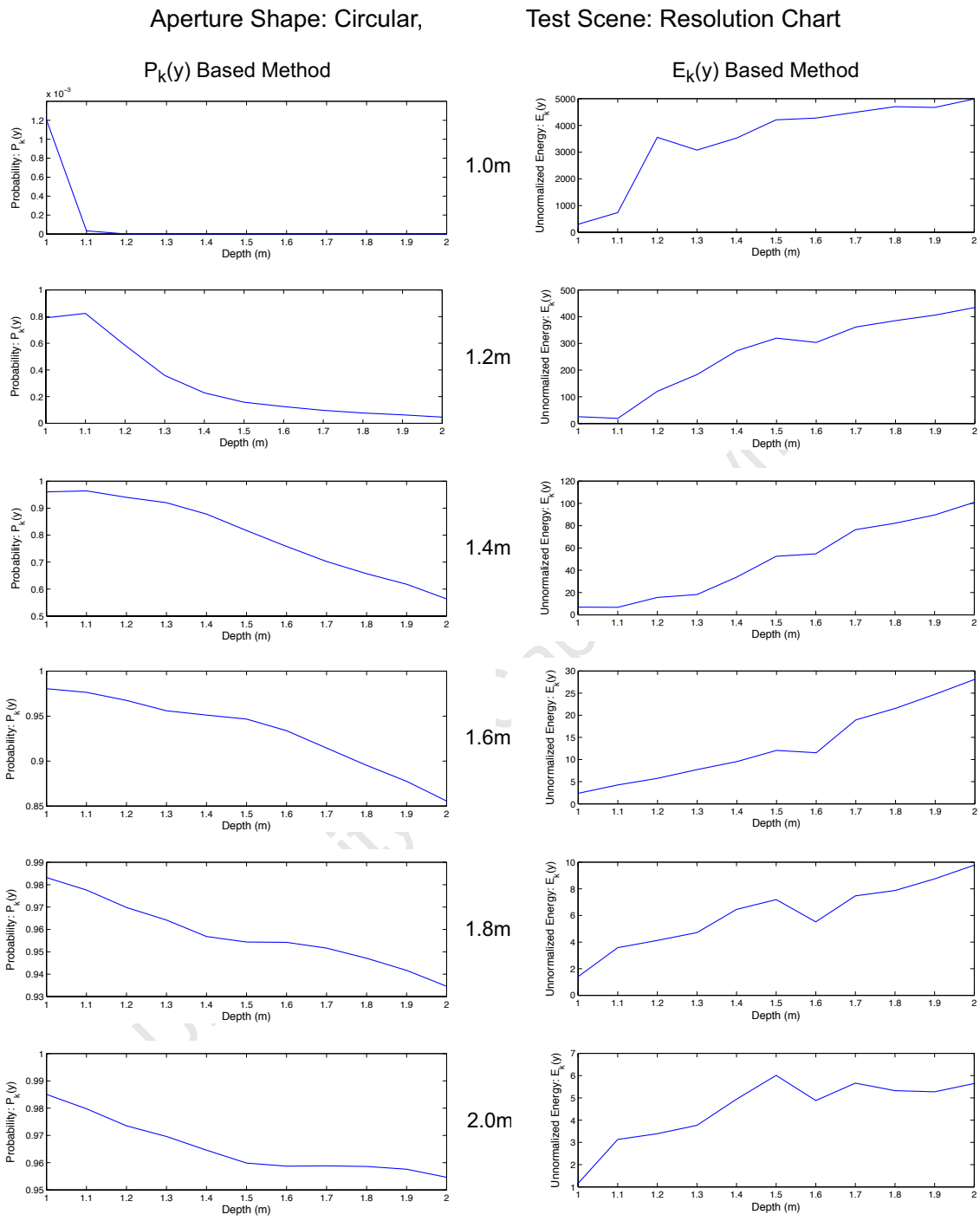



Figure 7.10: Diagram showing the results of depth estimation using a conventional circular aperture and the resolution chart scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

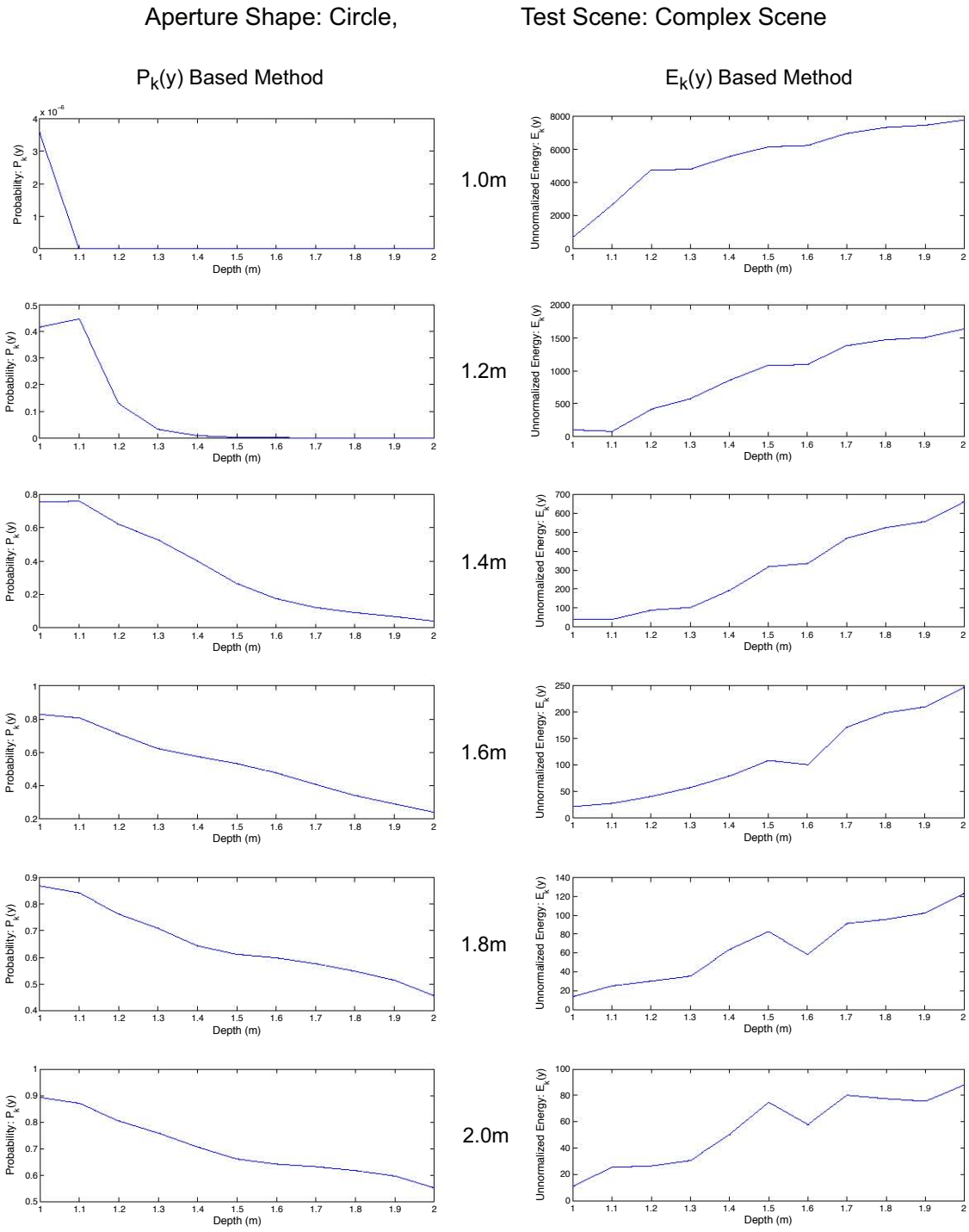



Figure 7.11: Diagram showing the results of depth estimation using a conventional circular aperture and the complex planar scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

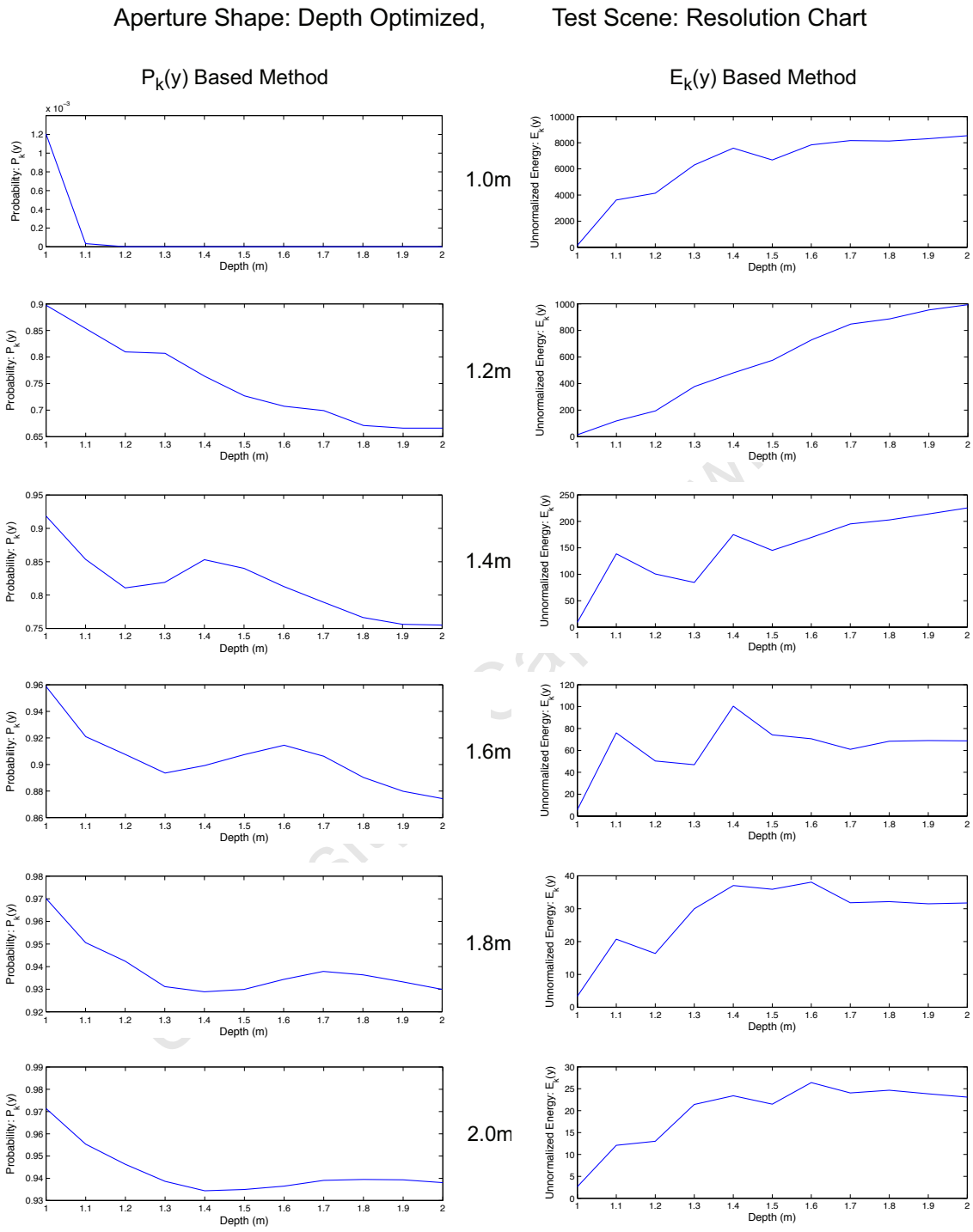



Figure 7.12: Diagram showing the results of depth estimation using the depth-optimized coded aperture and the resolution chart scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

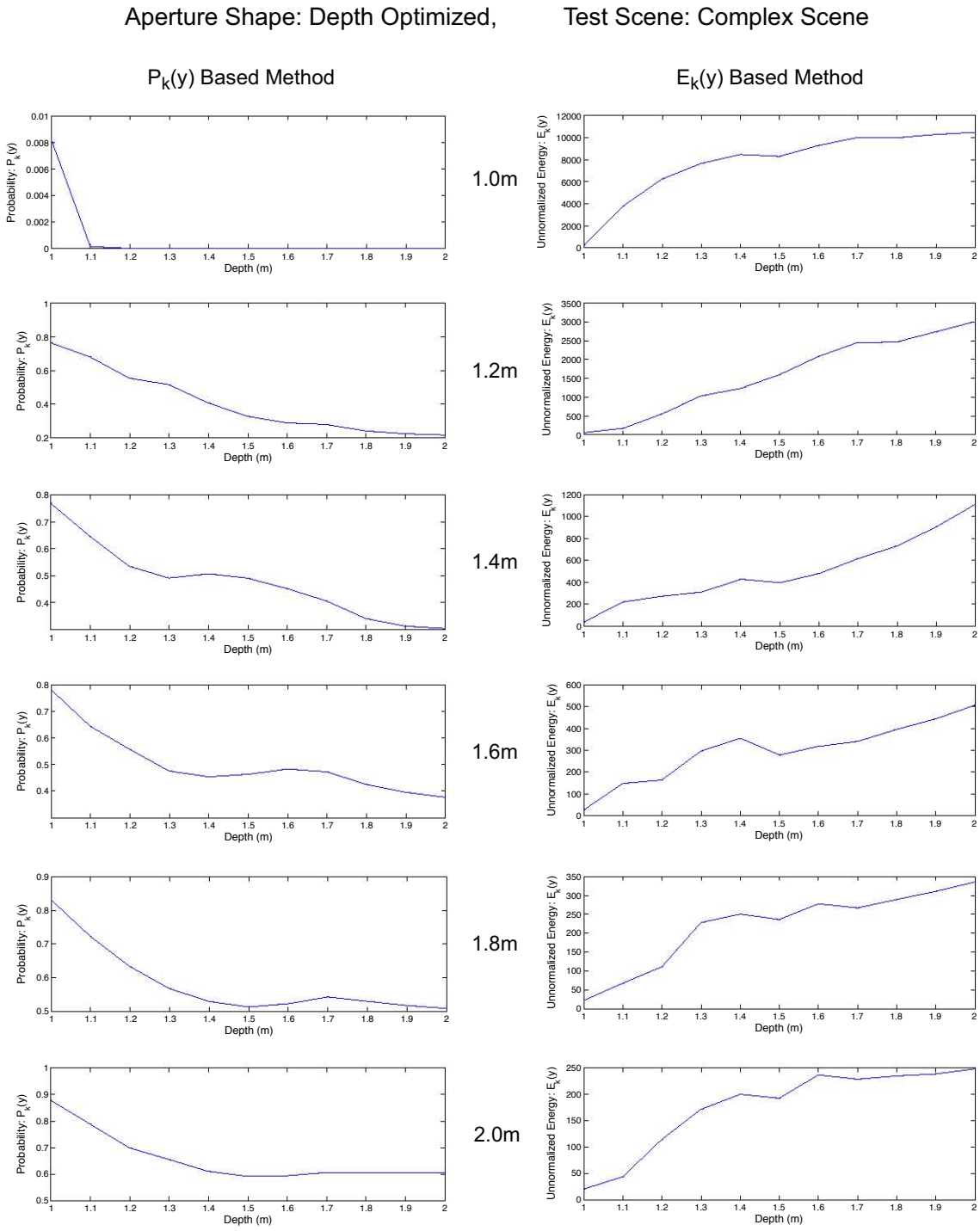



Figure 7.13: Diagram showing the results of depth estimation using the depth-optimized coded aperture and the complex planar scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

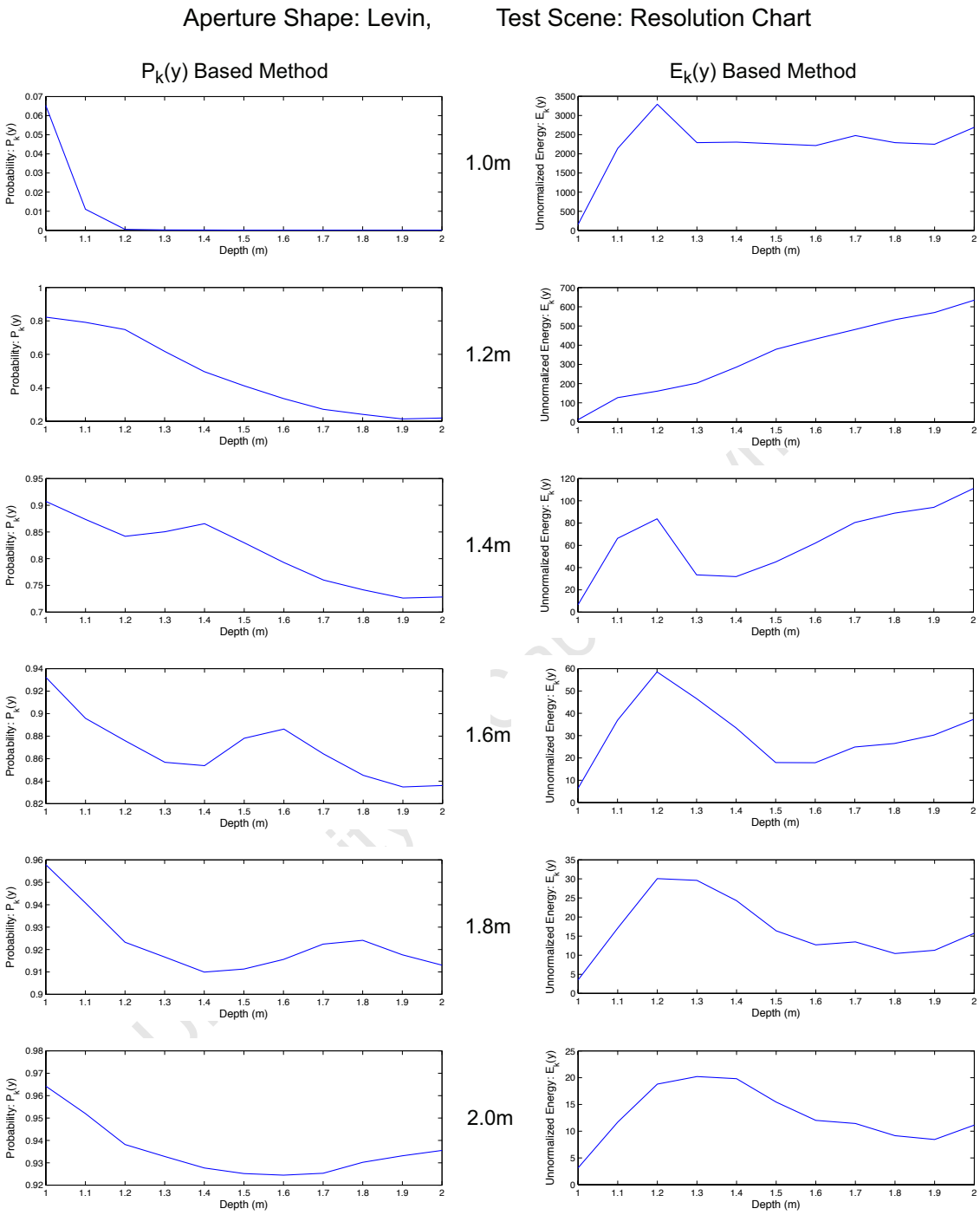



Figure 7.14: Diagram showing the results of depth estimation using Levin's coded aperture and the resolution chart scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

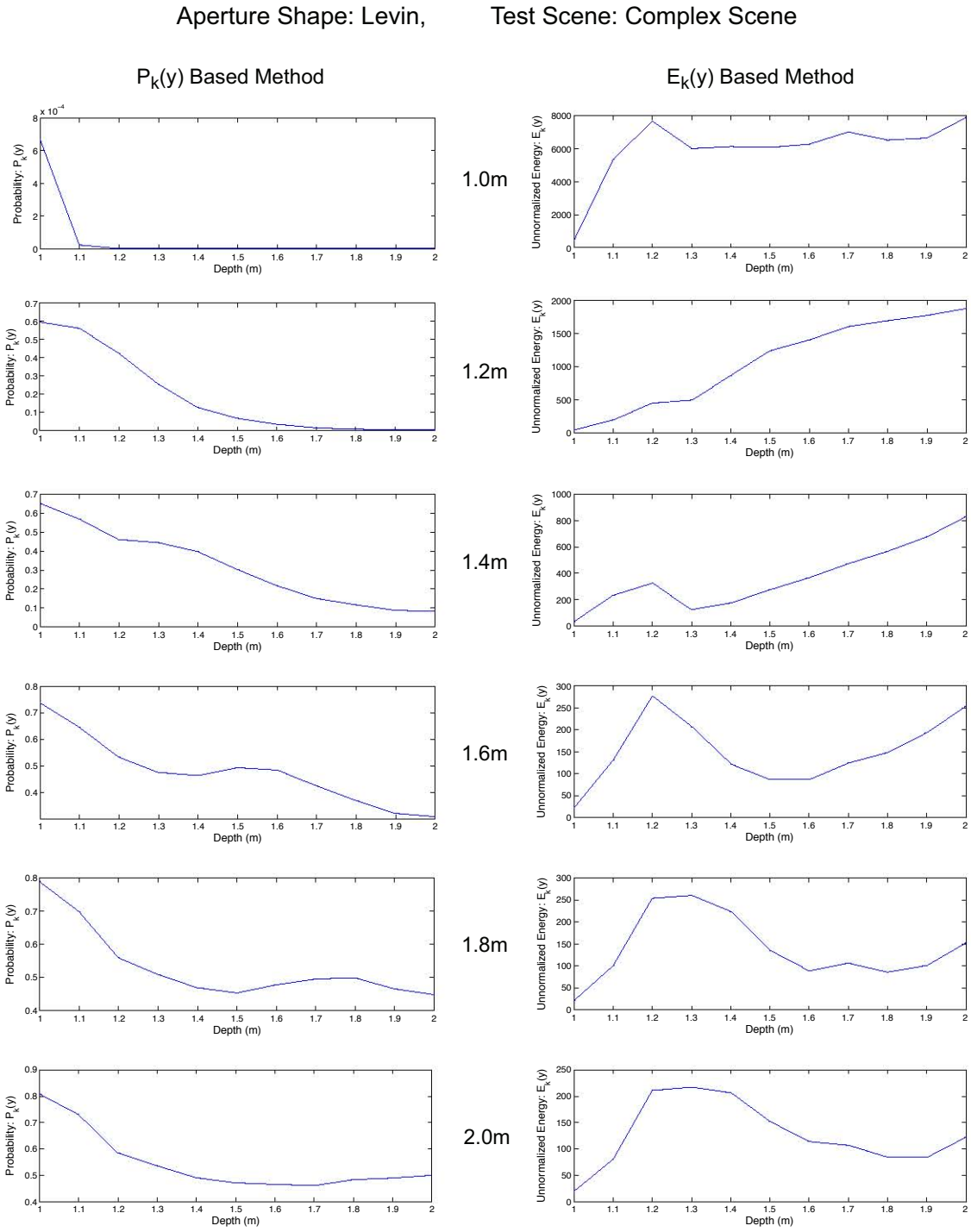



Figure 7.15: Diagram showing the results of depth estimation using Levin's coded aperture and the complex planar scene. Each row represents a different experimental scene-to-camera distance, ranging from 1.0m to 2.0m in 20cm increments. The left column shows the $P_k(y)$ probabilities for each possible distance, while the right column shows the unnormalized energies, $E_k(y)$.

## 7.2 Light Field Results

The simplest method of visualizing a captured light field is as a 2D array of 2D images. Figure 7.16 shows a subset of the light field captured in our experiment, in both uv-major indexing and xy-major indexing. These two visualizations are mathematically identical, and have been discussed in detail in sections 2.1 and 4.4. Under uv-major indexing, the $(u, v)$ array is of size $9 \times 9$, and each location in the array represents a unique $(x, y)$ image of size $1188 \times 792$ pixels. Each $(u, v)$ image is a unique view of the scene, as though the camera has been moved to a slightly different location. Under yx-major indexing, the $(x, y)$ array is of size $1188 \times 792$, and each location in the array represents a unique $(u, v)$ image of size $9 \times 9$. Each $(x, y)$ image represents a single pixel on the camera sensor, and each $(u, v)$ coordinate represents the intensity of light falling on the sensor pixel from a unique angular direction.

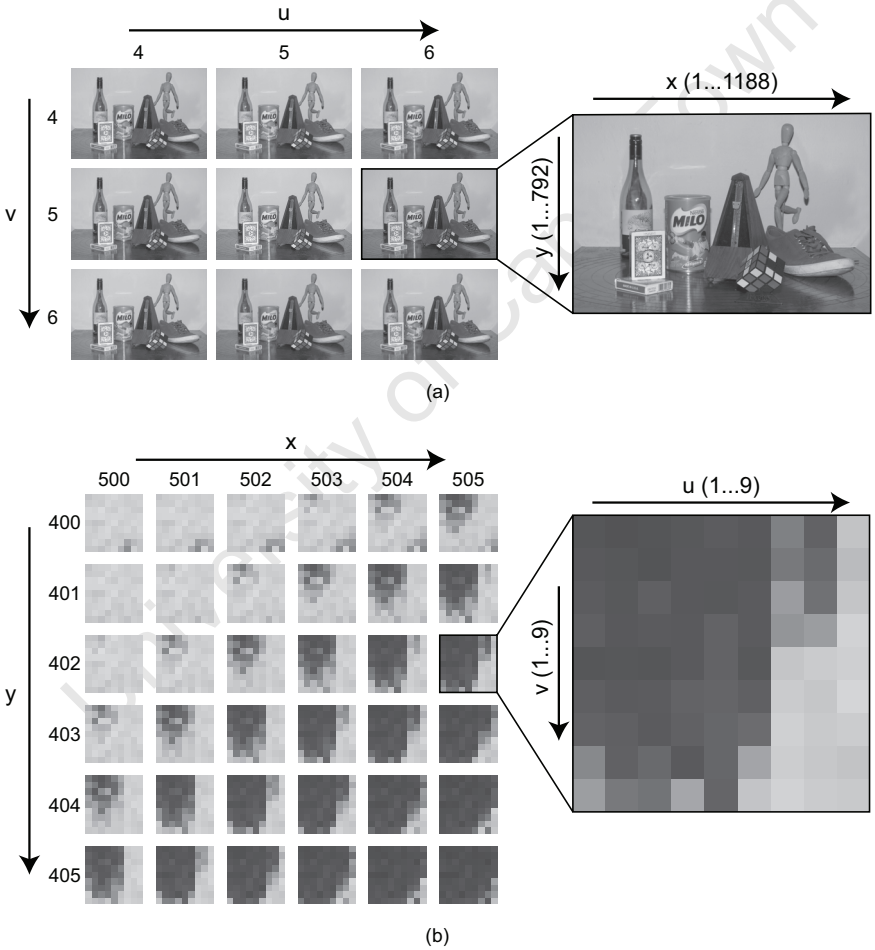
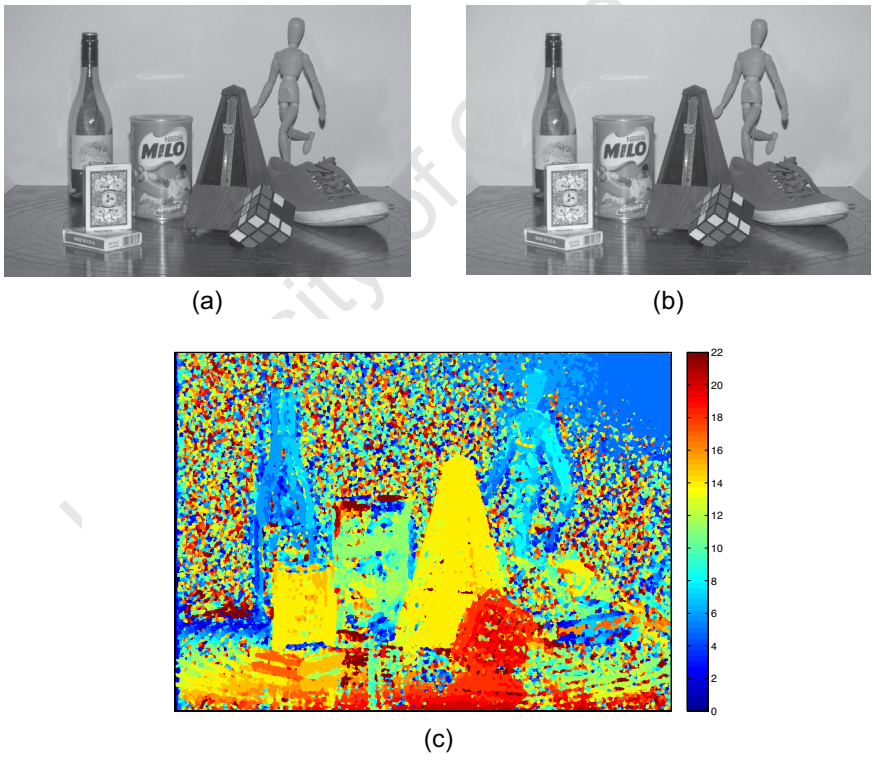



Figure 7.16: Diagram showing a subset of the captured light field in: (a) uv-major indexing and (b) xy-major indexing .

### 7.2.1 Stereo Pixel Disparity

Figure 7.17 (c) shows the results of performing a stereo disparity calculation on a pair of images extracted from the captured light field. The stereo input pair is shown in (a) and (b), and represents the images located at $(u, v) = (1, 5)$ and $(u, v) = (9, 5)$ respectively. The two images could be directly input into the stereo disparity algorithm without first being rectified because the two images have equal $v$ values and the $(u, v)$ axes are aligned with the sensor's $(x, y)$ axes.

Maximum pixel disparity corresponds to objects closest to the camera, and in our experiment the maximum value was 22 pixels. The stereo disparity algorithm functions by searching for matching blocks in the input image pair, and then calculates the change in horizontal position. To do this, each image block must have sufficient texture to make a reliable match. Therefore large, smooth areas in the input images, such as the white background in our test scene, cannot be expected to produce accurate results. If the noisy background values are ignored, the disparity calculation has actually performed quite well, especially taking into account the small change in view provided by the stationary camera.




(a) (b)

(c)

Figure 7.17: Diagram showing the stereo image pair (a & b) that was extracted from the light field, and the stereo disparity map (c) that was generated from it.

### 7.2.2 Synthesizing Virtual Photographs

#### Adjusting Aperture Shape

Figure 7.18 shows four virtual photographs calculated from the captured light field, using four arbitrary aperture shapes. The aperture in (a) is a centered transparent $9 \times 9$ block (which is equivalent to simply extracting the image located at $(u, v) = (5, 5)$ ), the aperture in (b) is fully open and so represents the image obtained when integrating the light field over all possible angular dimensions, and the apertures used in (c) and (d) are a diamond and a diagonal line respectively, and represent the images obtained when integrating the light field over a select subset of angular dimensions.

By integrating over a large set of angular dimensions the signal-to-noise ratio (SNR) of a photograph is improved, but this comes at the expense of decreasing the depth-of-field (i.e. the blur scale is increased for objects that lie outside the focal plane). The effect of the virtual apertures can clearly be seen in the cropped and magnified versions of the photographs, since the mannequin lies outside the focal plane. The single $(u, v)$ value used in (a) has produced the largest depth-of-field, and so the mannequin is hardly blurred at all, while the 81 $(u, v)$ values used in (b) has produced a significantly blurred image of the mannequin. The other two aperture shapes represent compromises between SNR and depth-of-field, where (c) produces a minor approximately circular blur, and (d) produces a unidirectional blur by integrating over a linear subset of angular dimensions.

#### Adjusting Focal Plane

Figures 7.19 – 7.21 (pages 117 – 119) show three virtually refocussed images calculated from the captured light field. The three images were selected out of the 41 refocussed images produced in the refocussing experiment, and correspond to the alpha values: $\alpha = 0.995$, $\alpha = 1.000$, and $\alpha = 1.005$ respectively. The full set of 41 refocussed images produced using alpha values ranging from 0.980 to 1.020 in 0.001 increments can be found electronically in appendix A.4. All the refocussed photographs were integrated over the entire set of $(u, v)$ values in order to reduce the field-of-view and thereby emphasize the effect of refocussing.

Figure 7.19 shows the photograph with its original focal plane ($\alpha = 1.000$) passing through the front face of the metronome in the middle of the scene. The cropped versions of the photograph show magnified views of three different objects in the scene, each located at a different depth. The Rubik's cube shown in (b) is located 20cm in front of the focal plane, (c) shows the metronome located on the focal plane, and (d) shows the head of the mannequin located 20cm behind the focal plane. Predictably, both the cube and the mannequin are slightly blurred, while the metronome is in sharp focus.

Figure 7.20 shows the photograph with its focal plane moved further away from the camera by placing the virtual image plane closer to the aperture plane ($\alpha = 0.995$). The success of the refocussing can clearly be seen in the cropped images shown in (b), (c), and (d). The mannequin's head, which was slightly blurred is now in sharp focus, while the originally sharp image of the metronome is now slightly blurred. The cube is now 40cm away from the new virtual focal plane, and therefore its image has become even more severely blurred.

Lastly, in figure 7.21 the photograph shown has had its focal plane moved closer to the camera than the original by placing the virtual image plane further away from the aperture plane ($\alpha = 1.005$). The new focal plane now passes through the cube, and so its image has become focussed and sharp. The originally sharp image of the metronome is now slightly blurred, and the image of the mannequin (which is 40cm away from the virtual focal plane) has become severely blurred.

One might be temped to think that the refocussing method described here is conceptually similar to the coded aperture deblurring shown in section 7.1. However, they are in fact fundamentally different. While coded aperture deblurring attempts to undo the effect of blurring after the angular dimensions of the light field have been integrated by the 2D sensor, the light field refocussing method requires capturing the entire 4D light field before its integration by the sensor. The extra two dimensions of information available to the light field method substantially simplifies the problem of defocus blurring by replacing the badly defined deconvolution step with a simple geometric optics model. This is the reason that the refocussing results shown in this section appear to be far superior to the results obtained in section 7.1. However, the superior refocussing results come at the expense of having to capture the full 4D light field just to produce a single 2D image. For many practical applications capturing the 4D light field (by sacrificing spatial or temporal resolution) may be infeasible, and in these situations deblurring using coded apertures might be the only option.

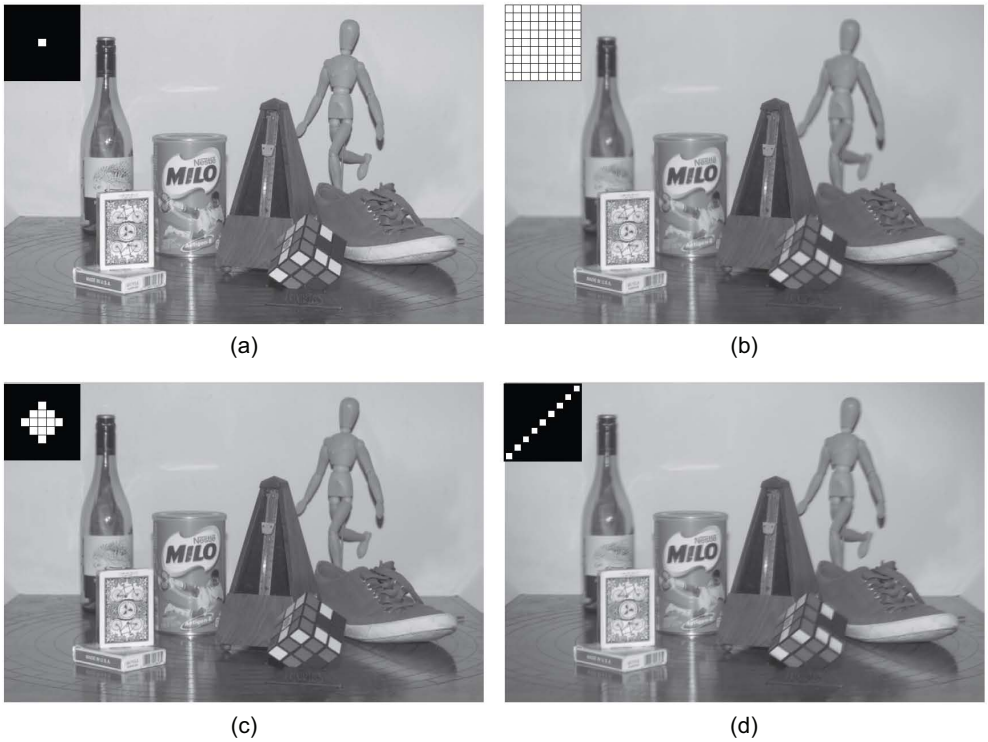
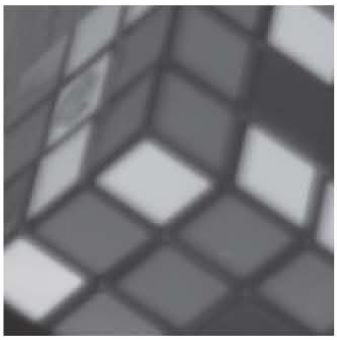
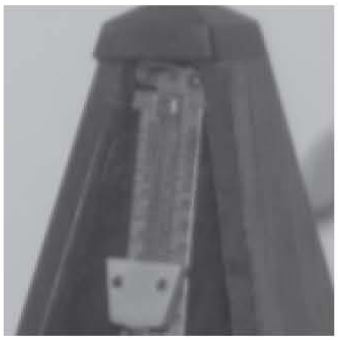



Figure 7.18: Figure showing the results of synthesizing virtual photographs with adjusted aperture shapes. The aperture shapes tested were: (a) a single centered binary block, (b) a fully open aperture, (c) a $3 \times 3$ diamond, and (d) a horizontal line.


(a)



(b)



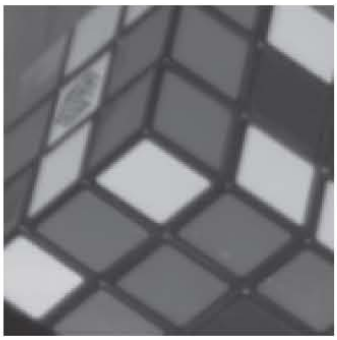
(c)

(d)

Figure 7.19: Figure showing the result of refocusing using the light field and a virtual image plane placed at $\alpha = 1.000$. (a) The full image, (b) crop of the Rubik's cube, (c) a crop of the metronome, (d) crop of the mannequin.


Figure 7.20: Figure showing the result of refocusing using the light field and a virtual image plane placed at $\alpha = 0.995$. (a) The full image, (b) crop of the Rubik's cube, (c) a crop of the metronome, (d) crop of the mannequin.


(a)



(b)

(c)

(d)

Figure 7.21: Figure showing the result of refocusing using the light field and a virtual image plane placed at $\alpha = 1.005$. (a) The full image, (b) crop of the Rubik's cube, (c) a crop of the metronome, (d) crop of the mannequin.

## 7.3 Coded Exposure Results

The results of the coded exposure experiments are shown in figures 7.22 – 7.25 (pages 122 – 125). Each figure shows the motion deblurring results obtained using a different exposure pattern, for both the planar resolution chart scene and the human face scene. Four exposure patterns were tested, namely: a conventional pulse, a coded pattern optimized using the fitness function $R_{raskar}(f_k)$, the pattern used by Raskar et al. in their coded exposure experiments [29], and a coded pattern optimized using the fitness function $R_{zhou}(f_k)$. The subtle details in the deblurred images contained in this section may not be easily visible due to their small size, but the full resolution images can be found electronically in appendix A.4.

Figure 7.22 shows the results of the experiment using a conventional pulse exposure. In the observed images there is a large, smooth vertical motion blur that that has hardly any high frequency information in the vertical direction. The text in the resolution chart is completely unrecognizable, the horizontal lines have become smeared over a number of pixels, and the vertical lines have merged into each other creating a complex indistinguishable pattern. In the human face scene, the elongated shape of the face can be seen, but no facial features can be identified. The structure of the collar is also no longer visible. In the deblurred images some of the vertical information has been strengthened, but most of the scene remains unrecovered, and background noise has been significantly amplified. The position of the text in the resolution chart has been recovered, but the characters themselves remain unrecognizable. The overall shape of the horizontal lines are now visible, but none of the individual lines can be distinguished and the vertical lines have become obscured by severe vertical artifacts. In the deblurred image of the face, the positions of some of the largest features (e.g. the forehead, chin and eyebrows) have been recovered, but the identity of the face remains unrecognizable.

Figures 7.23 and 7.24 show the results of deblurring using the coded exposure pattern optimized using $R_{raskar}$ and the pattern used by Raskar et al. in their experiments. The motion blur in the observed images is quite different from the blur obtained with the conventional exposure, in that it is less smooth and seems to have retained some vertical frequency information. The text in the chart is unrecognizable, and the horizontal lines exist as multiple overlapping vertical replicas. The elongated shape of the face is identifiable in the observed image, and some features such as the mustache are recognizable, but the identity of the face is not visible. However, in the deblurred images much of the original information has been salvaged. The text is now readable, and the medium-to-coarse horizontal lines can even be distinguished from each other. Even the hand at the top of the resolution chart has been recovered, and the thumb is clearly identifiable. In the deblurred images of the face, almost all of the major features have been recovered, and the identity of the face is now visible. While the background noise has been amplified and some vertical 'ghosting' is also

present, the deblurring results using these coded exposure patterns are clearly superior to the results obtained using the conventional exposure.

Lastly, in figure 7.25 the deblurring results obtained using the coded exposure pattern optimized by $R_{zhou}$ are shown. $R_{zhou}$ differs from $R_{raskar}$ in that it takes into consideration the amount of noise in the input image, as well as the statistical distribution of image power over frequency for natural images. Unsurprisingly the optimized exposure pattern differs quite markedly from the previous two patterns, most significantly in that there are far fewer state changes. The observed images show a motion blur that is more structured than the conventional exposure blur, but smoother than the blur produced by the previous two patterns. The text in the chart is again unrecognizable and the horizontal lines have three clear vertical replicas, which are assumed to be caused by the three distinct pulses in the exposure pattern. The deblurring has performed better than the conventional exposure pattern, since most of the textual characters can be identified and most of the facial features have been recovered. However, the recovered features have softer edges than in the deblurred images produced with the previous two coded apertures, and this suggests that less high frequency information was retained. It is also interesting to note that the deblurred images produced here contain significantly less background noise than in any of the other exposure patterns, and perhaps this is due to the fact that $R_{zhou}$ takes noise level into account when optimizing the exposure pattern.

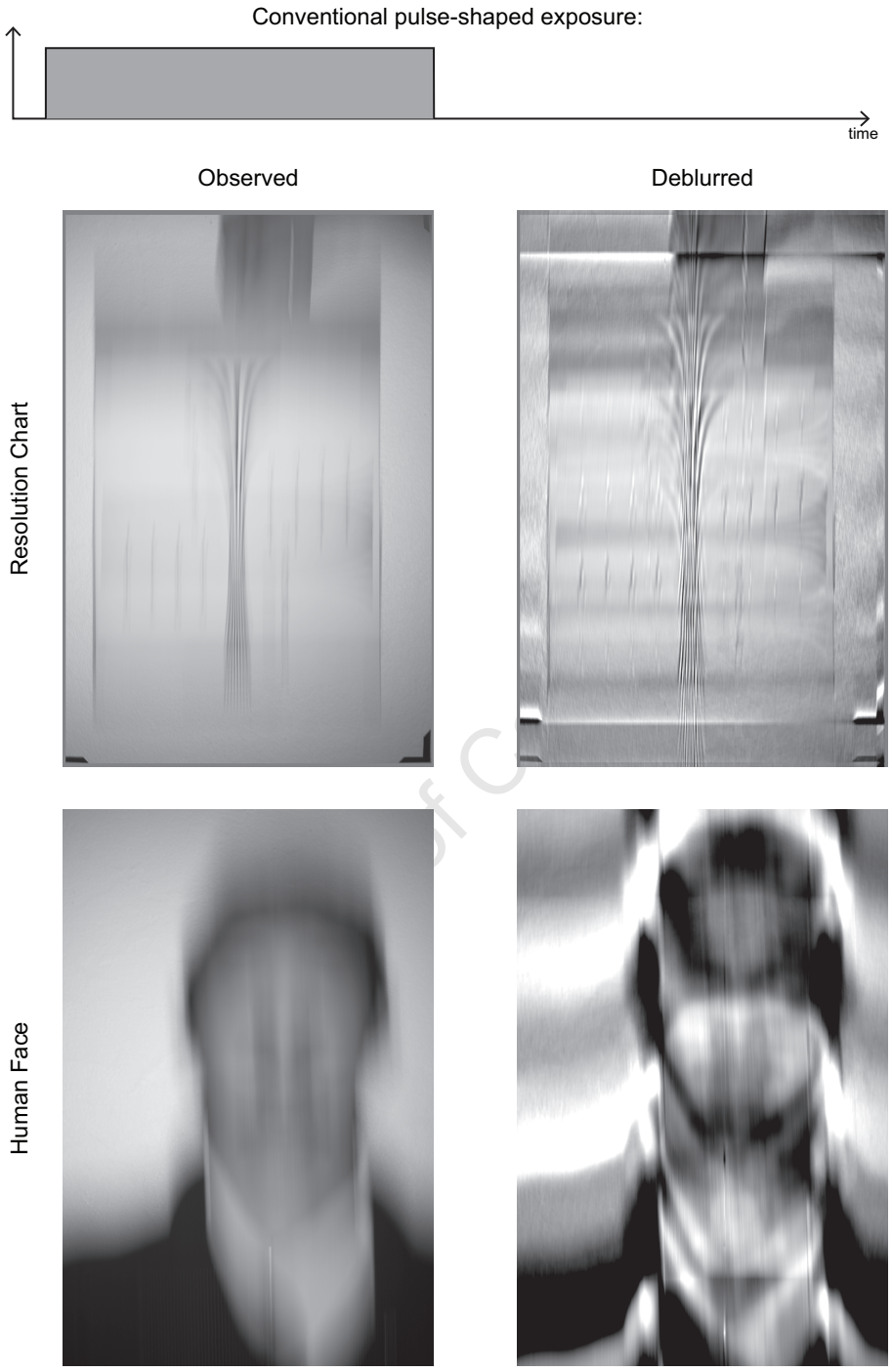



Figure 7.22: Diagram showing the results of motion deblurring using a conventional pulse-shaped exposure. Results are shown for both the resolution chart scene and the human face scene.

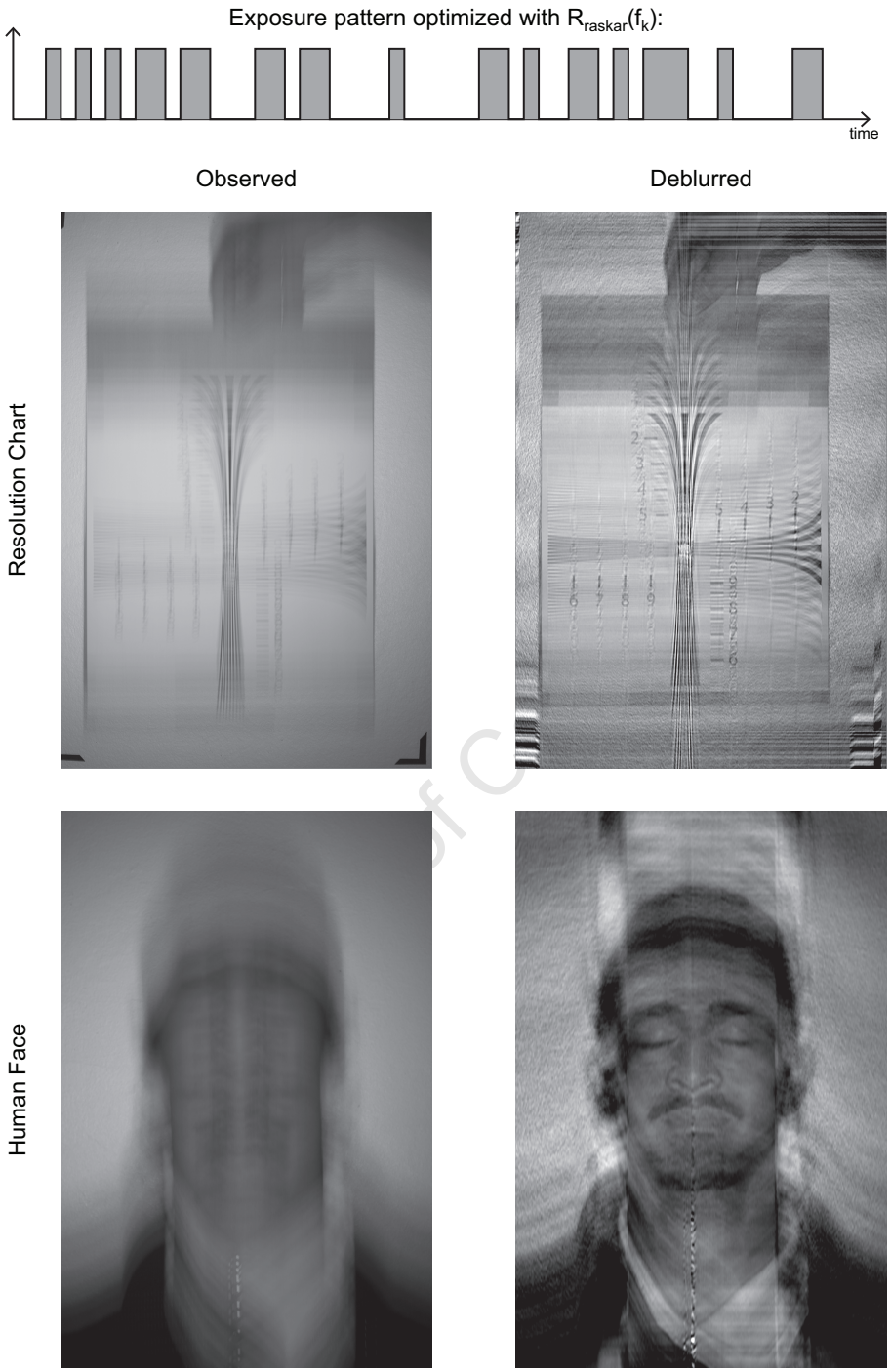



Figure 7.23: Diagram showing the results of motion deblurring using an exposure pattern that was optimized using the $R_{raskar}(f_k)$ performance metric. Results are shown for both the resolution chart scene and the human face scene.

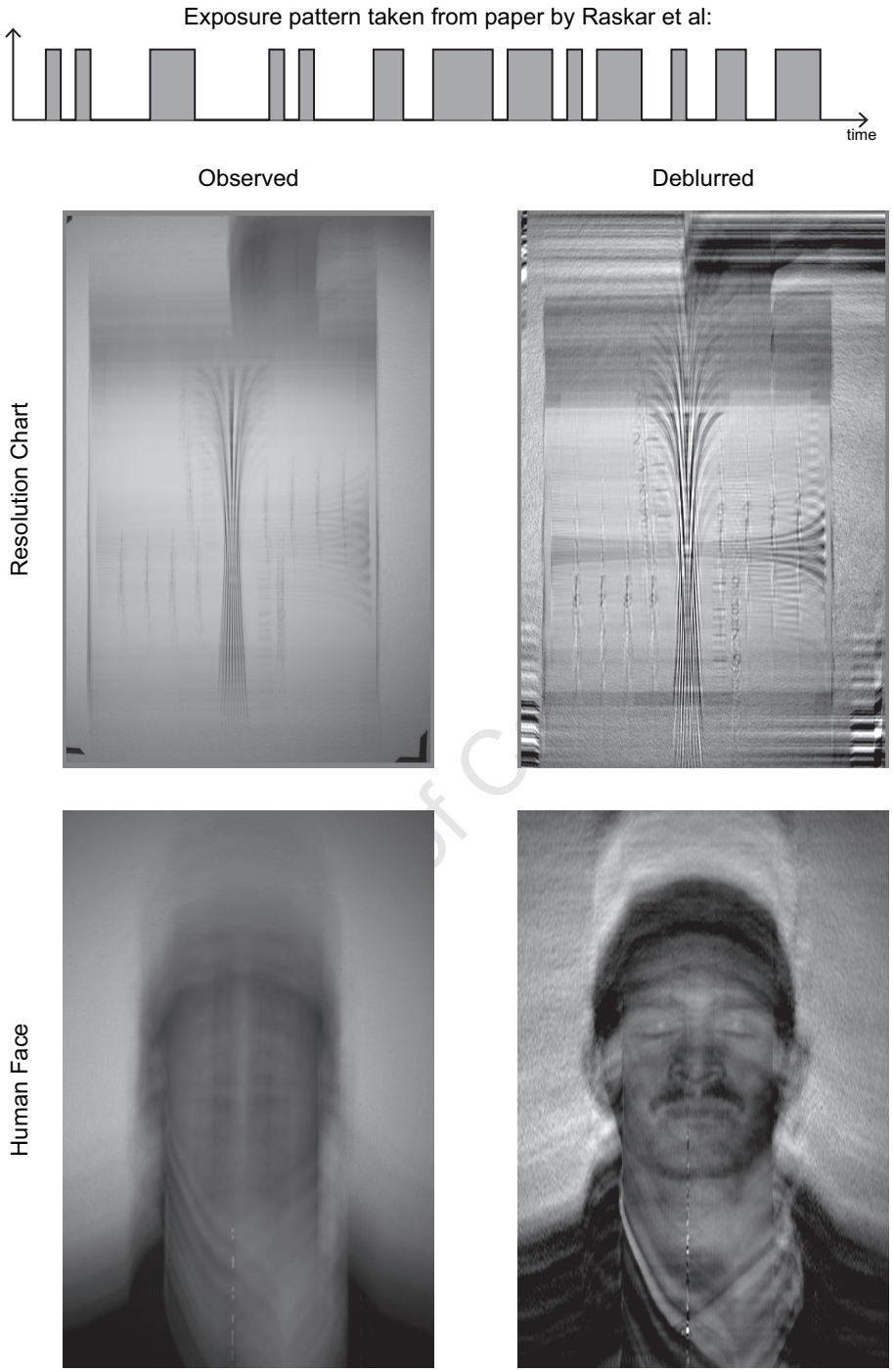



Figure 7.24: Diagram showing the results of motion deblurring using the exposure pattern developed by Raskar et al. [29]. Results are shown for both the resolution chart scene and the human face scene.

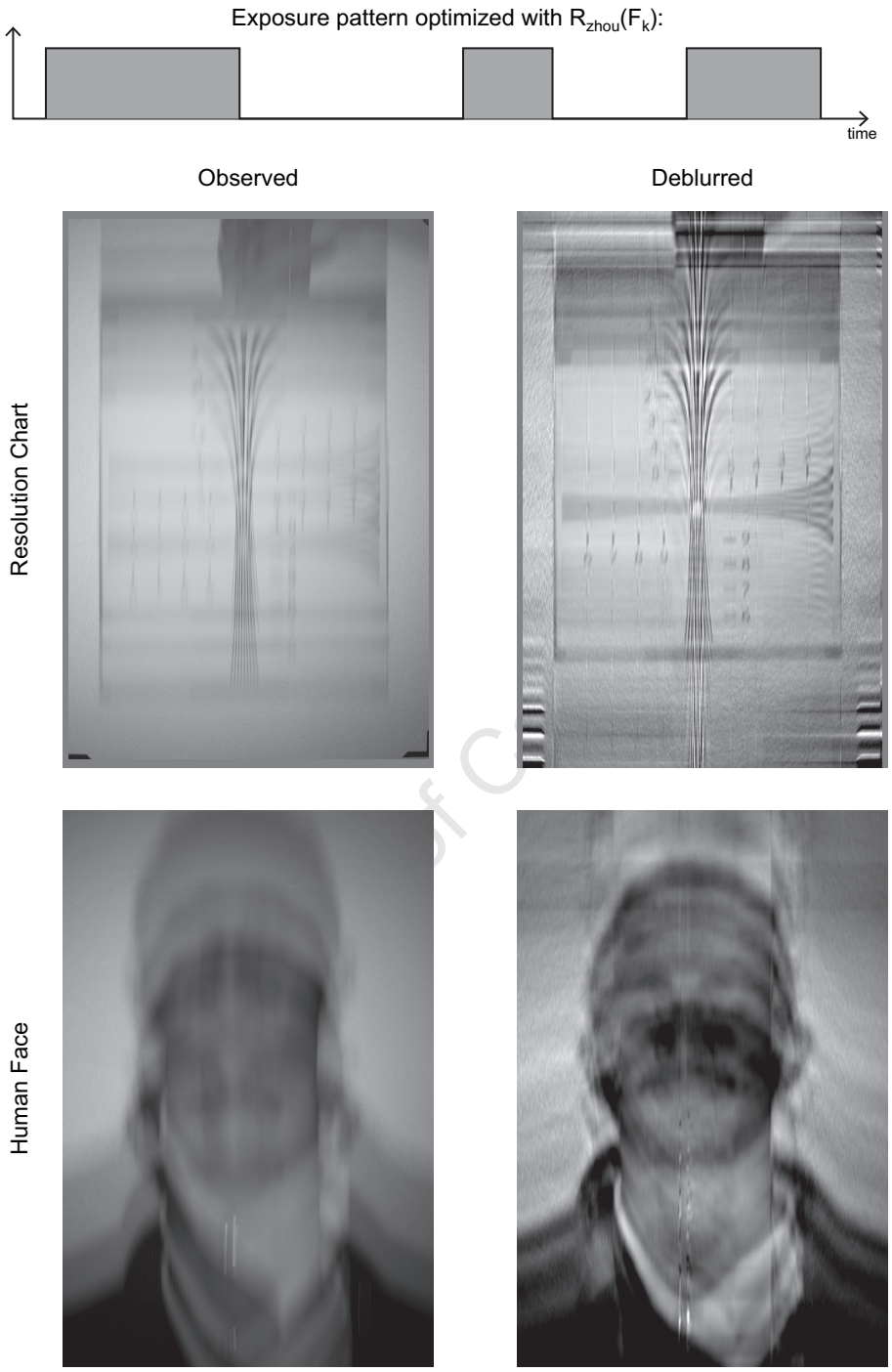



Figure 7.25: Diagram showing the results of motion deblurring using an exposure pattern that was optimized using the $R_{zhou}(F_k)$ performance metric. Results are shown for both the resolution chart scene and the human face scene.

# Chapter 8

# Conclusion

## 8.1 Summary of Results

This thesis is intended as an introduction to the field of computational photography, covering the basic objectives, theories and implementations that are currently being explored. It was also an opportunity to practically experiment with coded aperture and coded exposure techniques, and to determine their applicability to the tasks of defocus deblurring, motion deblurring, and depth estimation. The fulfillment of these objectives is summarized in the conclusions that follow.



**Coded Photography Prototype Camera**

In order to experiment with coded aperture and coded exposure techniques, a prototype coded photography camera was designed and constructed. The prototype camera consists entirely of inexpensive, off-the-shelf components, and where modifications were required, they were made with simple materials and tools. The completed prototype camera allows arbitrary aperture shapes to be inserted into the lens through slits cut into its outer housing. This means that the aperture shape can be quickly and easily replaced without disturbing the camera's orientation. The prototype camera also includes a reconfigurable illumination module, which can be used to illuminate a scene with an arbitrary binary illumination pattern, thereby encoding its exposure. Both of these modifications were tested, and they were found to function correctly.

**Coded Aperture Results**

The theory regarding the use of coded apertures for defocus deblurring and depth estimation has been discussed, and performance metrics were devised in order to design optimally performing apertures for both applications. The optimized apertures were then used to pho-

tograph test scenes at various depths, and the captured defocused photographs were input into the deblurring and depth-estimation algorithms. The output of these algorithms was analyzed, and the performances of the optimized apertures were compared to that of a conventional aperture and two other aperture shapes recommended by current literature.

In the deblurring experiments, the aperture shape optimized for refocusing performed just as well as the aperture recommended in the literature for deblurring. Additionally, it performed better than both the aperture optimized for depth estimation and the conventional circular aperture. This suggests that the method used for optimizing the aperture for defocus deblurring was successful.

In the depth estimation experiments, the aperture shape optimized for depth estimation performed better than the conventional circular aperture, but not quite as well as the aperture recommended in the literature for depth estimation. The most significant difference between our aperture and the recommended aperture was that the latter was symmetric, while ours was not. Therefore it is speculated that enforcing aperture symmetry may produce better depth estimation results.

**Light Field Results**

An introduction to the theory of light fields has been presented, as well as a brief overview of popular techniques for light field acquisition. One particular acquisition method uses multiple photographs taken with different coded aperture to capture the full 4D light field as a sequence of 2D slices. This method was described in detail, and the coded photography prototype camera was used to test the method. A sequence of 81 photographs was used to capture the light field of a test scene, and the result was shown as a 2D array of 2D images, in both uv-major indexing and xy-major indexing. The captured light field was also used for two practical applications, namely: stereo disparity calculation, and synthesizing virtual photographs.

Stereo disparity was calculated by extracting a stereo image pair from the light field (without requiring that the camera be moved), and inputting the image pair into a stereo disparity algorithm. The pair of images did not need to be rectified since they corresponded to the same $v$ value in the light field's $(u, v)$ coordinate system. Ignoring the noisy values produced for the white background, the result obtained from the stereo disparity algorithm is reasonably accurate, correctly identifying the relative depths of almost all the objects in the test scene. While calculating depth from a stereo image pair is not usually a noteworthy accomplishment, this stereo pair was taken from a single, stationary camera, and therefore represents a novel extension of the standard method.

Four virtual images were synthesized from the captured light field using the following virtual aperture shapes: a centered transparent binary block, a fully open aperture, a diamond, and a horizontal line. The results obtained clearly show the effect the aperture shapes have on the depth-of-field of the virtual photographs. Traditionally, using a large aperture increases the signal-to-noise ratio (SNR) but decreases the depth-of-field. However, by being able to generate virtual photographs with different aperture shapes allows the compromise between depth-of-field and SNR to be made after capture.

Refocussing experiments were also performed by synthesizing virtual photographs at a virtual image plane placed either closer to or further away from the aperture plane. A total of 41 images were produced, each with a slightly different focal plane, and three examples were provided in this report that show the successful refocussing of objects located outside the original plane of focus. The ability to adjust the position of the focal plane allows any object to be refocussed after capture, even when using a large aperture (and hence a shallow depth of field).

**Coded Exposure Results**

The theory regarding the formation of motion blur within a photograph has been presented, and the advantages of using a coded exposure for motion deblurring were explained. This theory was then used to formulate two exposure pattern performance metrics that could be used to design patterns specifically optimized for motion deblurring. The first metric is based on the related work by Raskar et al. [29], and the second is based on the metric used for optimizing apertures for defocus deblurring (which was originally based on related work by Zhou et al. [39]). Two exposure patterns were designed, each using one of the performance metrics, and they were then used to capture coded exposure photographs of moving test scenes. The captured images were then input into a deblurring algorithm and the results were analyzed in order to compare their performance.

Both of the optimized exposure patterns outperform the conventional pulse exposure in the deblurring experiments, since they are able to recover significant amounts of information from the observed blurry images. While the pattern based on Raskar et al.'s performance metric was able to recover more high frequency information than the pattern based on Zhou et al.'s metric, it also amplifies high frequency noise to a greater degree.

## 8.2 Recommendations for Further Research

### Coded Aperture Research

The coded apertures described in this report were hand-cut from cardboard strips and then inserted manually into the lens. This process is time consuming, and so the number of different aperture shapes that could be tested was limited. If the coded aperture could be implemented electronically (using an LCD array for example), the aperture could be instantly reconfigured. This opens up the possibility of calculating the optimal aperture shape based on a specific scene immediately before capturing the photograph. For example, the minima of the PSF's frequency spectrum could be engineered to coincide with the natural minima in the scene's frequency spectrum, thereby minimizing the loss of information. Using an electronic coded aperture together with a video camera would also allow each frame in the video to have a unique PSF.

Another way in which the coded aperture research could be extended is by using apertures constructed out of red, green and blue (RGB) filters instead of from simple opaque material. Since each filter would only allow light of a specific wavelength to pass through, the shape of an RGB aperture would be different for each color channel. Most current digital cameras contain a Bayer filter which only allows a single RGB band to land on each pixel. If the raw values could be extracted before interpolation, then each color channel would represent an image captured with a different PSF. If the three PSFs were significantly different, it is likely that where one had a minimum in its frequency spectrum the others would not, and in this way they could be used to approximate a singe PSF with a flat frequency spectrum.

Lastly, the results of the depth estimation experiments described in this report suggest that symmetric apertures perform significantly better than non-symmetric apertures. This postulate requires further experimentation in order to be confirmed.

### Light Field Research

When capturing a light field using coded apertures, the length of time required to take the multiple exposures (each with a unique aperture) is a significant limitation. Therefore this method would also benefit from replacing the manual coded aperture with an electronic coded aperture. For example, if an electronic aperture was placed within a video camera with a frame rate of 60fps, then a full light field (with an angular resolution of $9 \times 9$) could be captured in only 1.35 seconds.

**Coded Exposure Research**

The coded exposure experiments described in this report were performed by controlling the illumination of a scene within a darkened room, and while this produced adequate results, the method is restricted to use within a controlled laboratory environment. In real-world situations the ambient light in the environment cannot be controlled, and therefore in order to capture coded exposure photographs a fast programmable shutter must be used instead. Unfortunately mechanical shutters cannot operate at the high speeds required for coded exposures, so alternative methods need to be investigated such as using opto-chemical devices (e.g. an LCD filter).

Binary coded exposure patterns were chosen for our experiments due to their simplicity, but there is no reason to assume that binary patterns are optimal for motion deblurring. Therefore another avenue for further research would be to compare the results obtained using binary patterns with the results obtained using continuous or graduated exposure patterns.

# Bibliography


[1] Edward H. Adelson and James R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*, pages 3–20. MIT Press, 1991.

[2] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34 –47, nov/dec 2001.

[3] Moshe Ben-Ezra, Zhouchen Lin, Bennett Wilburn, and Wei Zhang. Penrose pixels for super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33:1370–1383, July 2011.

[4] P. Benzie, J. Watson, P. Surman, I. Rakkolainen, K. Hopf, H. Urey, V. Sainov, and C. von Kopylow. A survey of 3dtv displays: Techniques and technologies. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(11):1647 –1658, nov. 2007.

[5] Atmel Corporation. Atmel avr 8- and 32-bit microcontrollers - atmega328. `http://www.atmel.com/dyn/products/product_card.asp?part_id=4720`, November 2011.

[6] Paul Debevec, Ramesh Raskar, and Jack Tumblin. Computational photography: Advanced topics. ACM SIGGRAPH: Course Lecture Notes, 2008.

[7] N.A. Dodgson. Autostereoscopic 3d displays. *Computer*, 38(8):31 – 36, aug. 2005.

[8] Edward R. Dowski, Jr., Robert H. Cormack, and Scott D. Sarama. Wavefront coding: jointly optimized optical and digital imaging systems. In *SPIE Visual Information Processing IX*, pages 114–120, June 2000.

[9] Rob Fergus, Antonio Torralba, William T. Freeman, Rob Fergus, Antonio Torralba, and William T. Freeman. Random lens imaging. Technical report, MIT Computer Science and Artificial Intelligence Laboratory, 2006.

[10] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[11] Abhijeet Ghosh, Shruthi Achutha, Wolfgang Heidrich, and Matthew O'Toole. Brdf acquisition with basis illumination. *Computer Vision, IEEE International Conference on*, 0:1–8, 2007.

[12] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM.

[13] Martin Kemp and Margaret Walker, editors. *required*. Yale University Press, 1989.

[14] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 15–18, Washington, DC, USA, 2006. IEEE Computer Society.

[15] Sujit Kuthirummal and Shree K. Nayar. Multiview radial catadioptric imaging for scene capture. *ACM Trans. Graph.*, 25:916–923, July 2006.

[16] Bruce Lamond, Pieter Peers, and Paul Debevec. Fast image-based separation of diffuse and specular reflections. In *ACM SIGGRAPH 2007 sketches*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[17] Shawn Lankton. 3d vision with stereo disparity. `http://www.shawnlankton.com/2007/12/3d-vision-with-stereo-disparity/`, December 2007.

[18] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM.

[20] Jianguo Li, E. Li, Yurong Chen, Lin Xu, and Yimin Zhang. Bundled depth-map merging for multi-view stereo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2769 –2776, june 2010.

[21] Chia-Kai Liang, Tai-Hsu Lin, Bing-Yi Wong, Chi Liu, and Homer H. Chen. Programmable aperture photography: multiplexed light field acquisition. *ACM Trans. Graph.*, 27:55:1–55:10, August 2008.

[22] Francesc Moreno-Noguer, Peter N. Belhumeur, and Shree K. Nayar. Active refocusing of images and videos. *ACM Trans. Graph.*, 26, July 2007.

[23] Hajime Nagahara, Sujit Kuthirummal, Changyin Zhou, and Shree K. Nayar. Flexible depth of field photography. In *Proceedings of the 10th European Conference on Computer Vision: Part IV*, pages 60–73, Berlin, Heidelberg, 2008. Springer-Verlag.

[24] J. Nakamura. *Image sensors and signal processing for digital still cameras*. Optical Science and Engineering. Taylor & Francis, 2006.

[25] Shree K. Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graph.*, 25:935–944, July 2006.

[26] Shree K. Nayar and Srinivasa G. Narasimhan. Assorted pixels: multi-sampled imaging with structural models. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[27] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, Stanford University, April 2005.

[28] Y. Nomura, L. Zhang, and S.K. Nayar. Scene Collages and Flexible Camera Arrays. In *Proceedings of Eurographics Symposium on Rendering*, Jun 2007.

[29] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25:795–804, July 2006.

[30] Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[31] John A. Rogers, Heung Cho Ko, Mark P. Stoykovich, Jizhou Song, Viktor Malyarchuk, Won Mook Choi, Chang-Jae Yu, Joseph B. Geddes III, Jianliang Xiao, Shuodao Wang, and Yonggang Huang. A hemispherical electronic eye camera based on compressible silicon optoelectronics. *Nature*, 454:748–753, August 2008.

[32] Yoav Y. Schechner, Shree K. Nayar, and Peter N. Belhumeur. Multiplexing for optimal lighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:1339–1354, August 2007.

[33] Pradeep Sen, Billy Chen, Gaurav Garg, Stephen R. Marschner, Mark Horowitz, Marc Levoy, and Hendrik P. A. Lensch. Dual photography. *ACM Trans. Graph.*, 24:745–755, July 2005.

[34] Antonio Torralba and Aude Oliva. Statistics of natural image categories. In *Network: Computation in Neural Systems*, pages 391–412, 2003.

[35] Eric J. Tremblay, Ronald A. Stack, Rick L. Morrison, and Joseph E. Ford. Ultrathin cameras using annular folded optics. *Appl. Opt.*, 46(4):463–471, Feb 2007.

[36] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26, July 2007.

[37] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph.*, 24:756–764, July 2005.

[38] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24:765–776, July 2005.

[39] C. Zhou and S. K. Nayar. What are Good Apertures for Defocus Deblurring? In *IEEE International Conference on Computational Photography*, Apr 2009.

# Appendix A

# Electronic Resources

## A.1 Thesis Report as an Adobe Acrobat PDF

## A.2 Source Code

## A.3 Experimental Input Data

## A.4 Results


DVD Attached
If no DVD is attached here, then it
has been removed after submission.

# Appendix B

# Prototype Construction: Technical Documents



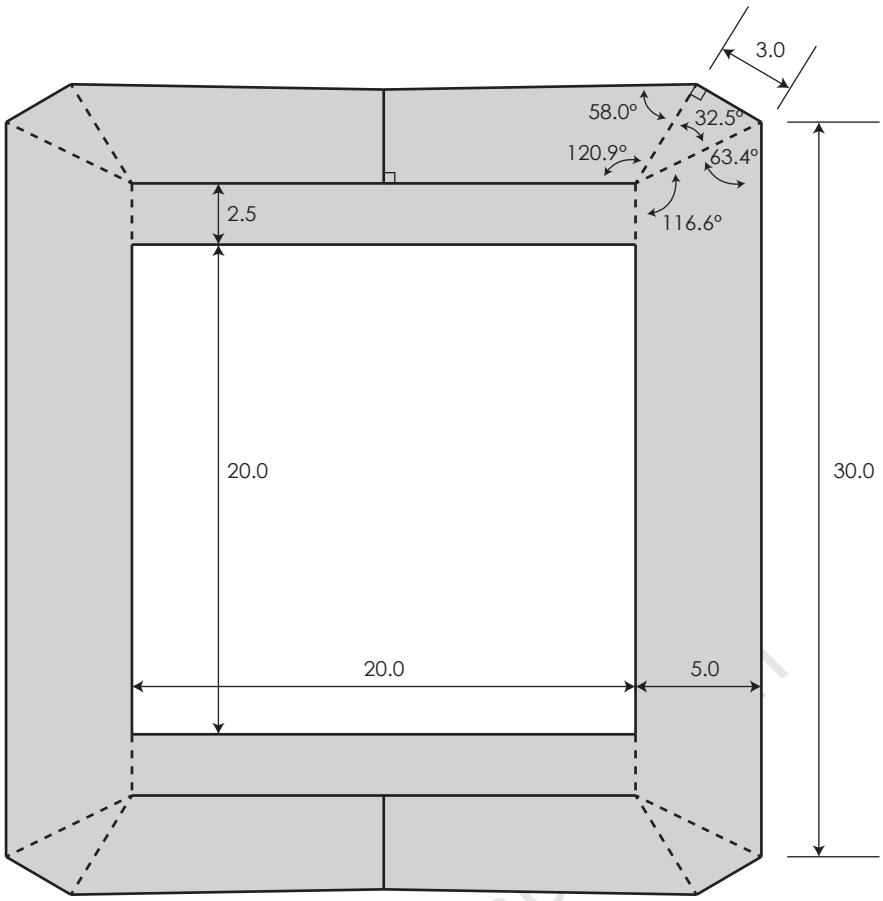
## B.1 Lens Disassembly Photographs


Figure B.1: Photograph showing lens disassembly. Step 1: Removal of front and rear caps, and removal of two micro-screws from mounting bracket.

Figure B.2: Photograph showing lens disassembly. Step 2: Removal of mounting bracket and side panel, and separation of inner and outer tubes.

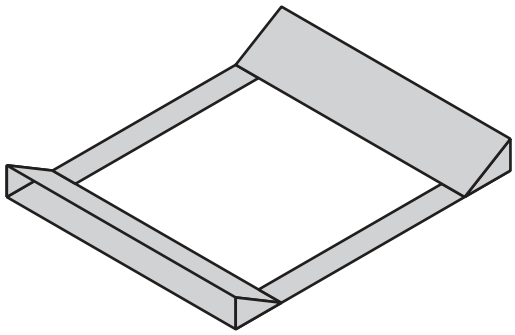

Figure B.3: Photograph showing lens disassembly. Step 3: Separation of inner-tube into focus adjustment ring, front optical element, and aperture plane.

Template Dimensions

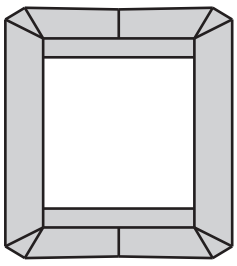



Scale: 3.33:1
All dimension in millimeters and degrees

Constructed Isometric



Cutting Template



Scale: 1:1

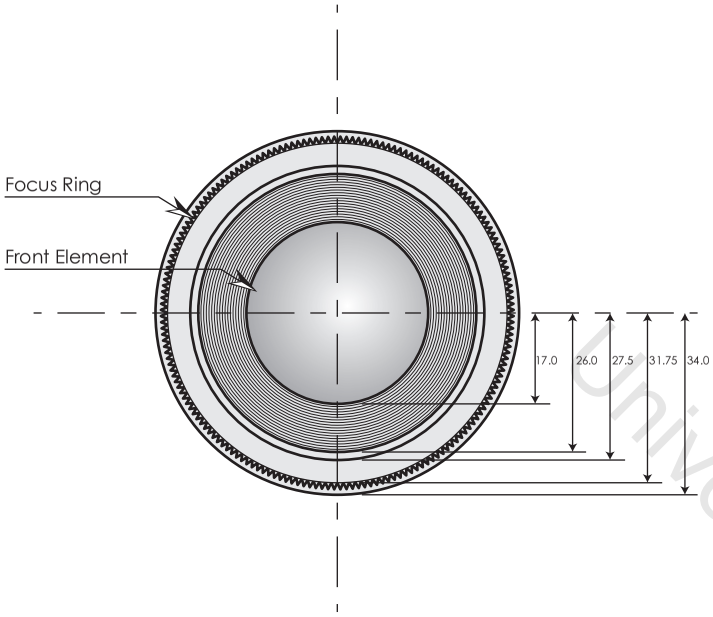


Martin Wilson
University of Cape Town
16 February 2011

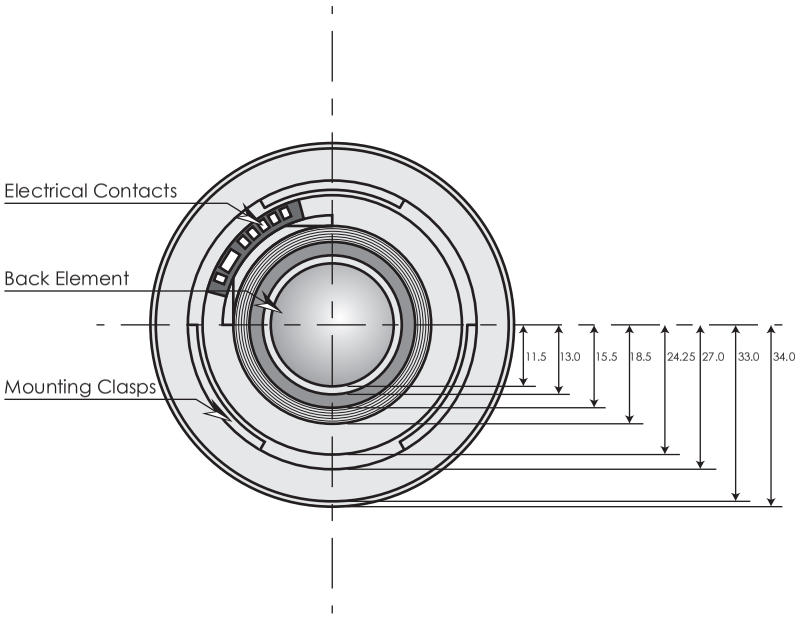
Internal Coded Aperture Guide Template

Rev 1

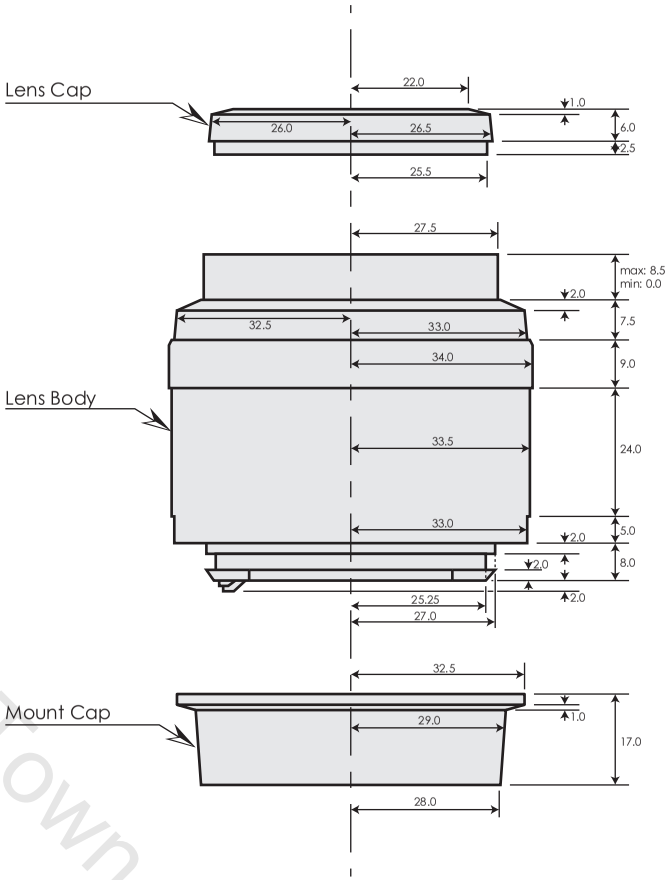
Top




Bottom




Side



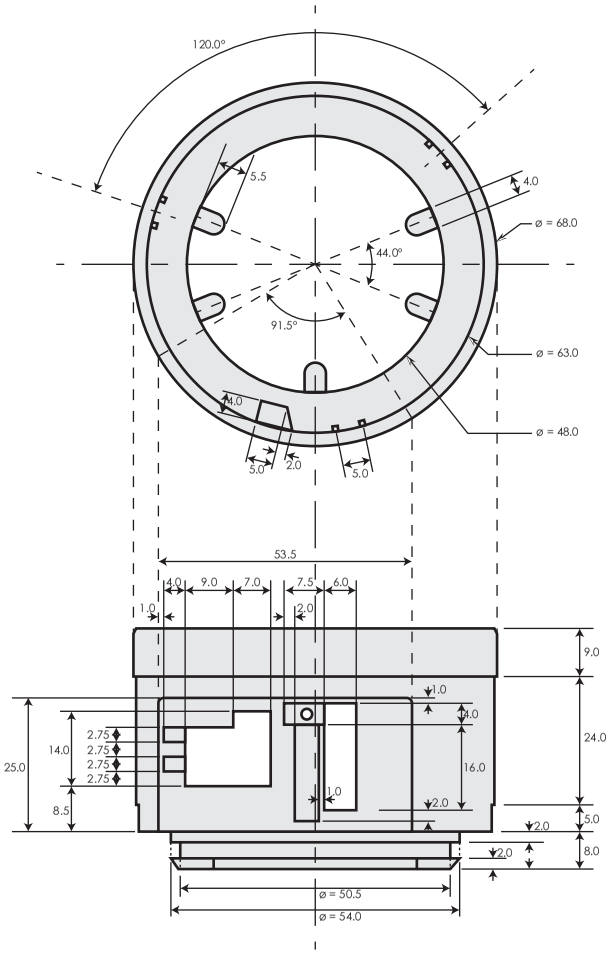



Martin Wilson | University of Cape Town | 10 November 2010

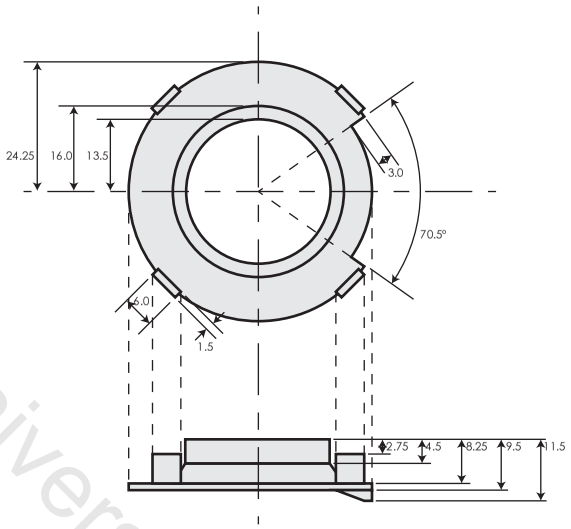
External Dimensions of Canon EF 50mm f/1.8 II Lens

Scale: 1:1.41 | All dimensions are in millimeters and degrees | Rev 1

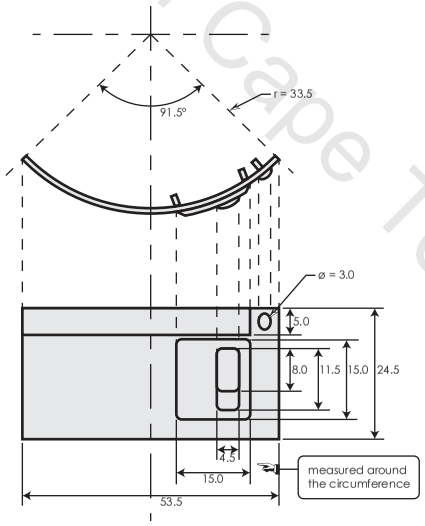
## Outer Housing Tube



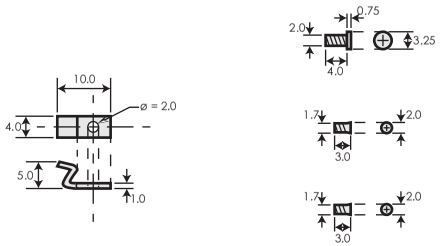

## Rear Mounting Cap



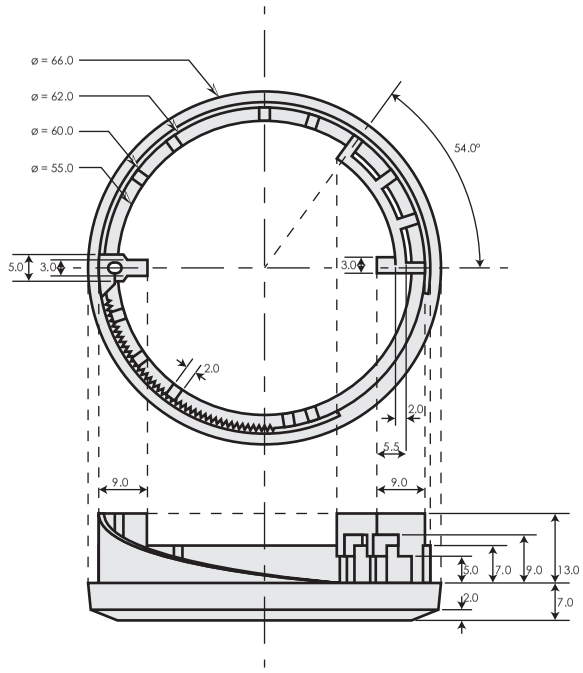

## AF/MF Selection Panel




## Screws and Fasteners




## Focus Adjustment Ring



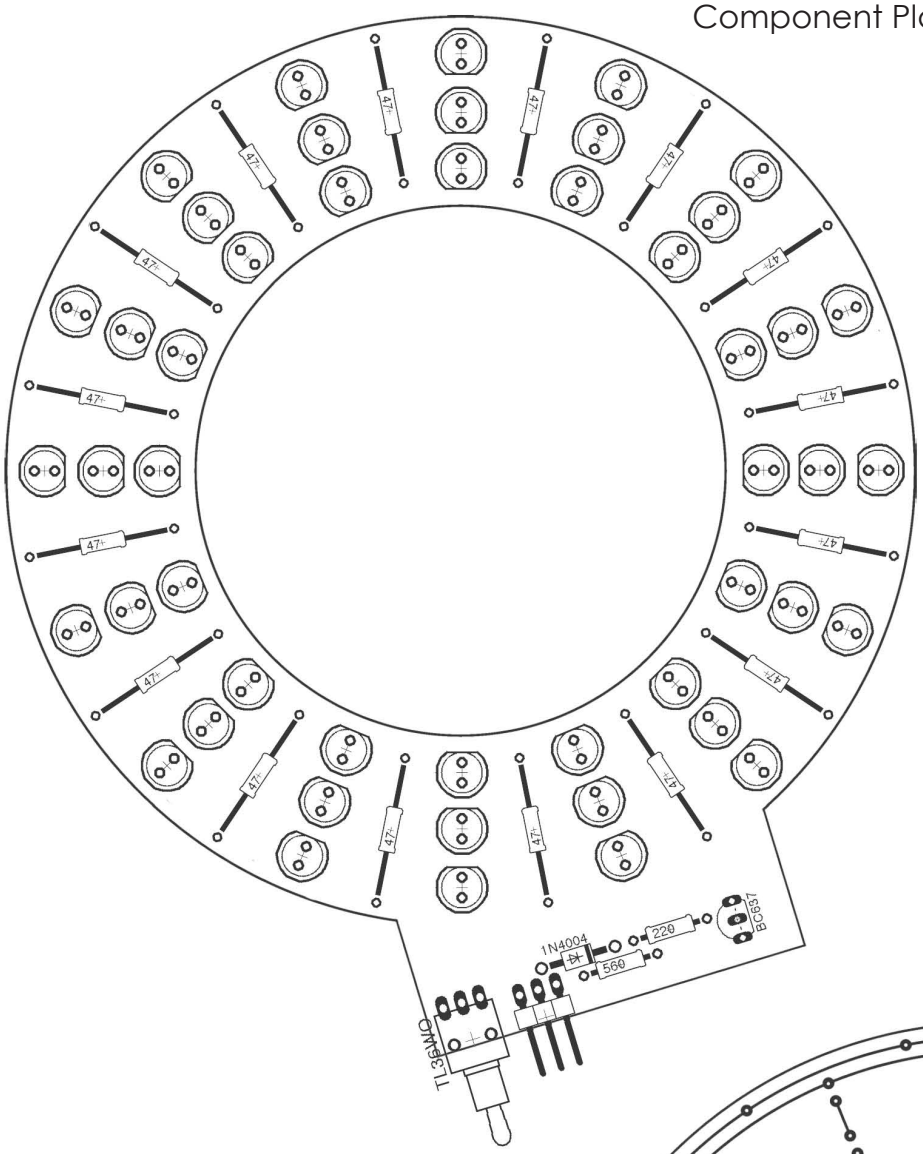



Martin Wilson — University of Cape Town — 16 November 2010

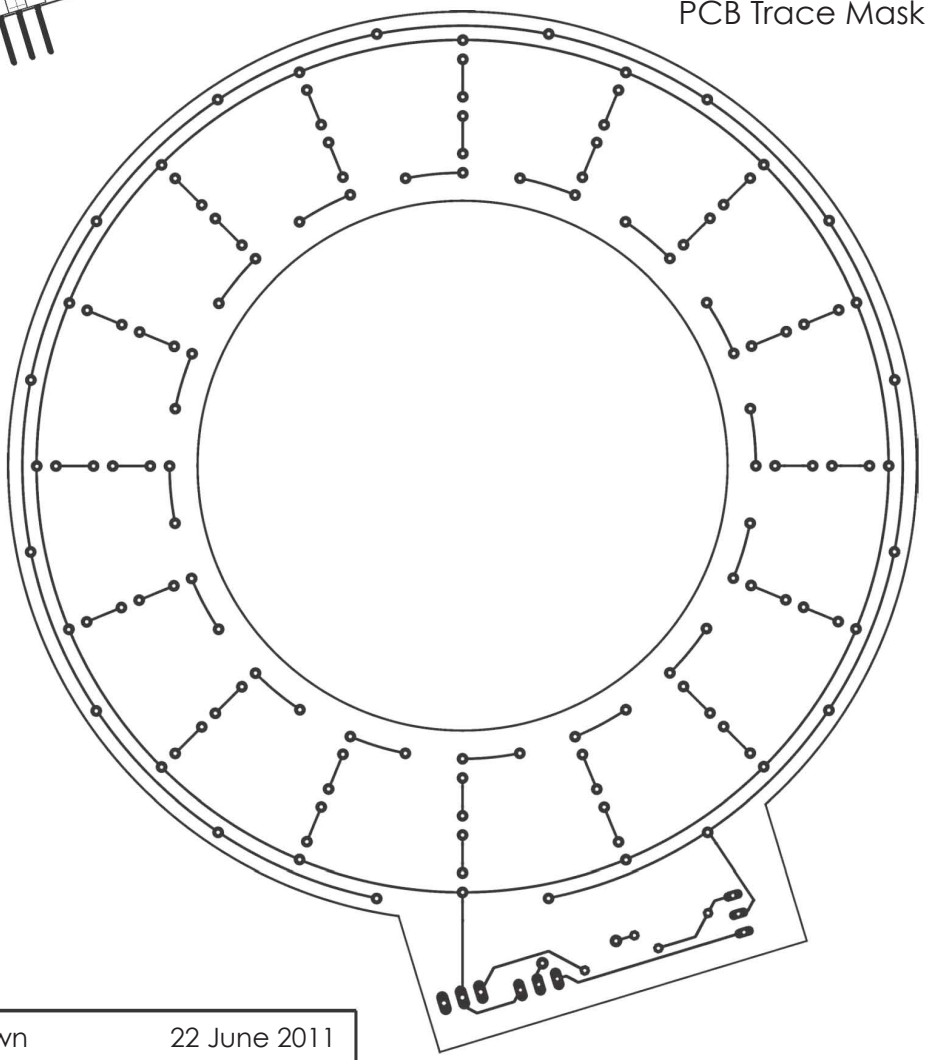
Internal Non-Optical Components of Canon EF 50mm f/1.8 II Lens

Scale: 1:1.14 — All dimensions are in millimeters and degrees — Rev 1

Component Placement




PCB Trace Mask



Martin Wilson University of Cape Town 22 June 2011

Circular LED Array: Printed Circuit Board

Rev 1

Scale 1:1

# EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department.
If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer - Zulpha.Geyer@uct.ac.za; Chemical Engineering Building, Upper Campus, UCT, (Ph 021 650 4791).
**NB: A copy of this completed form must be included with the thesis/dissertation/report when it is submitted for examination.**

---

Name of Principal Researcher/Student: Martin Wilson  Department: Electrical Engineering

Preferred email address of applicant: Martin.Wilson@uct.ac.za

If a Student:  Degree: MSc  Supervisor: Dr Fred Nicolls

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: Coded Aperture and Coded Exposure Photography

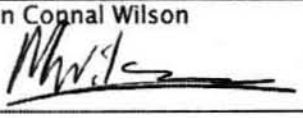
**Overview of ethics issues in your research project:**

| | | |
|---|---|---|
| **Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?** | YES | (NO) |
| **Question 2: Is your research making use of human subjects as sources of data?** If your answer is YES, please complete Addendum 2. | (YES) | NO |
| **Question 3: Does your research involve the participation of or provision of services to communities?** If your answer is YES, please complete Addendum 3. | YES | (NO) |
| **Question 4: If your research is sponsored, is there any potential for conflicts of interest?** If your answer is YES, please complete Addendum 4. | YES | (NO) |

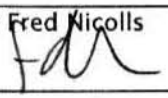
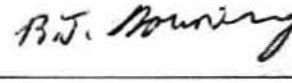
If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

**I hereby undertake to carry out my research in such a way that**

- there is no apparent legal objection to the nature or the method of research; and
- the research will not compromise staff or students or the other responsibilities of the University;
- the stated objective will be achieved, and the findings will have a high degree of validity;
- limitations and alternative interpretations will be considered;
- the findings could be subject to peer review and publicly available; and
- I will comply with the conventions of copyright and avoid any practice that would constitute plagiarism.

| Signed by: | Full name and signature | Date |
|---|---|---|
| Principal Researcher/Student: | Martin Connal Wilson  | 16/09/2011 |

This application is approved by:

| | | |
|---|---|---|
| Supervisor (if applicable): | Dr Fred Nicolls  | 16/09/2011 |
| HOD (or delegated nominee): Final authority for all assessments with NO to all questions and for all undergraduate research. |  | 16/9/2011 |
| Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above questions. |  | 24/10/2011 |

**ADDENDUM 2:** To be completed if you answered YES to Question 2:

It is assumed that you have read the UCT Code for Research involving Human Subjects (available at http://web.uct.ac.za/depts/educate/download/uctcodeforresearchinvolvinghumansubjects.pdf) in order to be able to answer the questions in this addendum.

| | | |
|---|---|---|
| 2.1 Does the research discriminate against participation by individuals, or differentiate between participants, on the grounds of gender, race or ethnic group, age range, religion, income, handicap, illness or any similar classification? | YES | (NO) |
| 2.2 Does the research require the participation of socially or physically vulnerable people (children, aged, disabled, etc) or legally restricted groups? | YES | (NO) |
| 2.3 Will you not be able to secure the informed consent of all participants in the research? (In the case of children, will you not be able to obtain the consent of their guardians or parents?) | YES | (NO) |
| 2.4 Will any confidential data be collected or will identifiable records of individuals be kept? | YES | (NO) |
| 2.5 In reporting on this research is there any possibility that you will not be able to keep the identities of the individuals involved anonymous? | (YES) | NO |
| 2.6 Are there any foreseeable risks of physical, psychological or social harm to participants that might occur in the course of the research? | YES | (NO) |
| 2.7 Does the research include making payments or giving gifts to any participants? | YES | (NO) |

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

Photographs of the individuals' faces must be included in any reporting of this research, as the ability to recover the identity of a human face from a blurred photograph is one of the research goals.

In order to protect the anonymity of the individuals, only images of their faces will be kept on record. None of the following information will be kept: name, age, race, sex, address, contact details. However, the chance remains that the individuals will be able to be recognised from the photographs. This will be clearly explained to all individuals who take part in the research, and they will be allowed to choose between giving their permission, or opting-out entirely.

# INFORMATION SHEET & CONSENT FORM
## (for participants being photographed)

Coded Aperture and Coded Exposure Photography:
an investigation into applications and methods.

Hello, my name is Martin Wilson, and I am conducting research towards a master's degree at the University of Cape Town. I am researching the use of coded apertures and exposures for reducing defocus and motion blur in photographs, and I would like to invite you to participate in the project.

**Project Overview**
Digital cameras currently use circular apertures and 'box' exposures; however these traditional techniques are not ideal since they lose visual information in the form of defocus and motion blurring. Theoretically, specially designed aperture shapes and exposure patterns should be able to preserve more information than the traditional approaches and can therefore be used to reduce the amount of blur in a photograph. I am interested in testing this hypothesis by capturing blurry photographs of test scenes (including human faces) with a modified camera, and attempting to remove the blur.

**Clarification of Participation**
Please understand that you do not have to participate (i.e. your participation is voluntary). The choice to participate is yours alone. If you choose not to participate, there will be no negative consequence. If you choose to participate, but wish to withdraw at any time, you will be free to do so without negative consequence. However, I would be grateful if you would assist me by allowing me to photograph you.

Should you choose to participate, you will be asked to pose in standing and sitting positions while I photograph your head and shoulders from various distances and using various camera settings. A total of 110 photographs will be required, and it should take no more than 45 minutes from start to finish. The photographs will be taken in an indoor studio environment, and flash illumination will be used throughout. I am not offering any compensation (monetary or otherwise), and your participation will have no direct benefit to yourself.

Only the photographs will be used in my research, and no personal information (e.g. name, age, sex, contact details, etc.) will be stored at any time. The photographs will not be sold, broadcasted or used for any other purposes, however they may be used in academic publications in order to display the results of my research. Your facial features will be clearly visible in all the photographs, and therefore I cannot guarantee your anonymity. After the completion of my research the photographs will be securely archived for a period not longer than two years, after which time they will be deleted.

Thank you for considering participating in my research project.

I _____ (print full name) with ID number _____, hereby acknowledge that I have read and understood the attached “INFORMATION SHEET & CONSENT FORM (for participants being photographed)” regarding participation in the research project entitled “Coded Aperture and Coded Exposure Photography”. Furthermore, I give consent for my photograph to be taken, used for research purposes, and published in the manner described in the afore-mentioned document. The decision was my own and was made voluntarily, without coercion from the researcher.

Sign: _____, at _____

Date: _____